

Amazing Computing™

Volume 1 Number 9
U.S.A. \$3.50
Canada \$4.50

Commodore Amiga™ Information and Programs



The background image shows a person from the chest up, wearing a grey pinstriped suit jacket and a dark red tie with white polka dots. They are pulling the jacket open with both hands, revealing a white t-shirt underneath. On the t-shirt is the Commodore Amiga logo, which consists of a stylized 'A' made of four parallel diagonal lines in red, orange, and yellow, followed by the word 'AMIGA' in a bold, black, serif font.

AMIGA

*Doing Business
with your Amiga™*

*Reviews:
TDI-Modula-2
Gizmoz
Dos 2 Dos
Maxi Plan*

*Special!
Amazing Directory*

*Sources and Resources for your
Commodore Amiga™*

Superbase™ PERSONAL RELATIONAL DATABASE SYSTEM

SUPERBASE IS NOW AVAILABLE FOR THE



The enormously popular and proven database system, Superbase is now available for the Amiga computer. We completely rewrote Superbase to take full advantage of all the power available on the amazing Amiga. This is not a conversion, but an entirely new Amiga program!

TOTAL SOLUTIONS

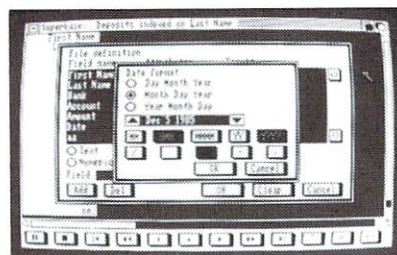
Superbase provides the total information management solution. It is a true productivity program for the Amiga computer. You can finally use a serious database with a serious machine. It's easy to keep track of inventories for your business whether you're working with parts inventory or real estate listings. Superbase is perfect for church membership rolls, patient files, personnel schedules or any place you need to manage and control large amounts of important information.

Access the power of the first **true** relational database management system with Superbase Personal Relational Database System. It will turn your Amiga into a truly productive tool, with virtually limitless capacities. Imagine being able to have an unlimited number of files open at any time. You can even have each file indexed with up to 999 key fields to search and sort.

EASY TO SET UP

Superbase utilizes the latest ideas in easy-to-use mouse and windowing technology. There are pull-down help menus to ease you through problems that may occur during database creation. Superbase is completely menu driven and takes advantage of the point-to-click features possible with the Amiga mouse.

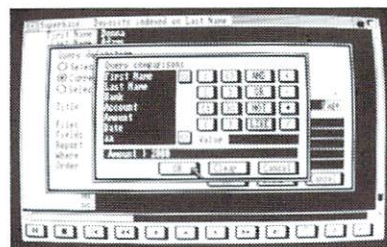
Create a database in minutes using the easy to understand menu selections and control panels. Type in field names, add details like length or data style and you are quickly ready to input your data. Unlike other databases, you can alter your formats at anytime, **without disturbing** the data already in existing files. Using Superbase's Enhanced BASIC, your database can be totally customized to virtually any application.



IT'S EASY TO MANAGE YOUR DATA

Display your data in the format you choose. Either page by page or just as it appears in the record format. You choose how to view the data you need. There is practically no limit to the number of fields in a record, you have complete control over what is displayed on screen or printed in custom reports.

Decide on the fields and on the sequence, then use the VCR style controls to view your data -- Get the first record, then fast forward, pause, continue or stop -- it's as easy as playing a video tape!



WORKING POWER

Superbase makes it easy to define reports or generate relational queries across multiple files, with multiple sort levels if you need them. Import data from other databases or applications. Export data to your favorite word processor, or join several files to form a new database.

The advanced B+ tree file structure and disk buffering means high performance -- Superbase reads a typical name and address record in an incredible three hundredths of a second!

THE VIDEO DATABASE

Superbase includes an amazing array of data types in your record format, including the names of pictures or digitized images stored on disk. Read the words, then look at digitized pictures you have already stored on disk. Your data records can "point" to images to recall them for viewing!

You can even link multiple images to a single database record to run automatic slide shows. It's all easily done using the VCR style commands that you control. Revise, update or review your illustrated database in any desired arrangement. You have total control! Superbase is the total software solution for people who must manage information.

THE BEST HAS ARRIVED!

Finally, a program that utilizes ALL the power and functions of the Amiga computer. Superbase brings to the Amiga the business solutions you have been waiting for.

The power of Superbase is also available for the Commodore 64/128 and the Apple IIe/IIc.

WHEN
QUALITY COUNTS!

PROGRESSIVE
PERIPHERALS
& SOFTWARE

464 KALAMATH STREET
DENVER, COLORADO 80204
303-825-4144
TELEX: 888837

Superbase Personal, (Amiga, Commodore 64/128), Apple IIe/IIc, are registered trademarks of Precision Software Ltd., Commodore Business Machines, Apple Computers, respectively. This ad and all of its contents are copyrighted by Progressive Peripherals & Software, Inc. and may not be reproduced, or duplicated in any manner without written permission.

Amazing Computing™

MetaScope: The Debugger

MetaScope gives you everything you've always wanted in an application program debugger:

- **Memory Windows**
Move through memory, display data or disassembled code live, freeze to preserve display and allow restoration.
- **Other Windows**
Status windows show register contents and program state with freeze and restore; symbol, hunk, and breakpoint windows list current definitions.
- **Execution Control**
Breakpoints with repetition counts and conditional expressions; trace for all instructions or subroutine level, both single-step and continuous execution.
- **Full Symbolic Capability**
Read symbols from files, define new ones, use anywhere.
- **Powerful Expression Evaluation**
Use extended operator set including relationals, all assembler number formats.
- **Direct to Memory Assembler**
Enter instruction statements for direct conversion to code in memory
- **and More!**
Mouse support for value selection and command menus, log file for operations and displays, modify/search/fill memory, etc.

MetaTools I

A comprehensive set of tools to aid your programming (full C source included):

- **Make**
Program maintenance utility.
- **Grep**
Sophisticated pattern matcher.
- **Diff**
Source file compare.
- **Filter**
Text file filter.
- **Comp**
Simple file compare.
- **Dump**
File dump utility.
- **Whereis**
File locator utility.

MetaScribe: The Editor

MetaScribe has the features you need in a program editor:

- **Full Mouse Support**
Use for text selection, command menus, scrolling — or use key equivalents when more convenient.
- **Multiple Undo**
Undo all text alterations, one at a time, to level limited only by available memory.
- **Sophisticated Search/Replace**
Regular expressions, forward/backward, full file or marked block.
- **Multiple Windows**
Work with different files or different portions of the same file at one time.
- **Macro Programs**
Lisp-like macro language lets you customize and extend the editor to meet your needs.
- **Virtual Memory**
Set the amount of data memory to be used, transparently edit files larger than memory.
- **and More!**
Keystroke macros for repetitive text, copy between files, block copy/paste/delete, set tabs and margins, etc.

DosDisk

A program that lets you access PC-DOS/MS-DOS™ diskettes on your Amiga. Use it to list file information and copy files between the PC-DOS/MS-DOS diskettes and Amiga diskettes or devices. Patterns can be used for file names, and you can even operate on all files in a directory at one time. A copy option converts source file line-end sequences as the copy is performed.

Metadigm, Inc.

19762 MacArthur Blvd.
Suite 300
Irvine, CA 92715
(714) 955-2555

MetaScope
\$95.00

MetaScribe
\$85.00

MetaTools

\$69.95

DosDisk

\$49.95

(California residents add 6% sales tax).
Visa/MasterCard accepted.

Dealer Inquiries Welcome

Amiga is a trademark of Commodore-Amiga Inc.
MS-DOS is a trademark of Microsoft, Incorporated

Metadigm products are designed to fully utilize the capabilities of the Amiga™ in helping you develop your programs. If you're programming the Amiga, you can't afford to be without them.

Publisher: Joyce Hicks
Circulation Manager: Doris Gamble
Assistant to the Publisher:
Robert James Hicks
Corporate Advisor: Robert Gamble

Managing Editor: Don Hicks
Hardware Editor: Ernest P. Viveiros
Amicus & Technical Editor: John Foust
Music Editor: Richard Rae
Art Director: Keith Conforti
Assistant Editor: Ernest P. Viveiros Jr.
Assistant Advertising Manager:
John David Fastino
Production Manager: Mark Thibault

Amazing Authors
Ervin Bobo
Bryan Catley
John Foust
Don Hicks
Kelly Kauffman
Perry Kivolowitz
George Musser Jr.
Steven Pietrowicz
Rick Wirch
&
The Amiga

Special Thanks to:
Robert H. Bergwall
RESCO, Inc.
E.P.V. Consulting
New England Technical Services
Interactive Tutorials Inc.

Advertising Sales
&
Editorial

1-617-678-4200

Amazing Computing™ (ISSN 0886-9480) is published by PIM Publications, Inc. P.O. Box 869, Fall River, MA. 02722. Subscriptions: in the U.S. 12 Issues for \$24.00; Canada and Mexico, \$30.00; Overseas, \$35.00. Printed in the U.S.A. Copyright© 1986 by PIM Publications, Inc. All rights reserved.

First Class or Air Mail rates available upon request.

PIM Publications, Inc. maintains the right to refuse any advertising

Amazing Computing™

Amazing Contents

Volume 1 Number 9

Making Cents in Business with your Amiga™ reviews by Ernest P. Viveiros	9
Dos 2 Dos reviewed by Richard Knepper Transfer files from PC/MS-DOS and AmigaBasic	13
MaxiPlan reviewed by Richard Knepper The Amiga version of Lotus 1-2-3	17
Gizmoz by reviewed by Peter Wayner	21
The Loan Information Program by Brian Catley A basic program to "review" your options	25
Starting Your Own Amiga Related Business by William Simpson The possible ways to establish your business.	33
Keep Track of Your Business Usage for Taxes by James Kummer A program to justify your Amiga to the IRS	37
Instant Music reviewed by Steve Pietrowicz	45
Mindwalker reviewed by Richard Knepper	47
The Alegra Memory Board reviewed by Rick Wirth	49
TxEd reviewed by Jan and Cliff Kent	51
Amazing Directory A guide to the sources and resources of the Amiga	53
Amiga Developers A listing of Suppliers and Developers	66
Public Domain Software Catalog The complete list of current AMICUS and Fred Fish PDS	69
The Absoft Amiga Fortran Compiler reviewed by Richard A. Reale	77
Using Fonts from AmigaBasic, Part Two by Tim Jones The Amiga Basic program	81
AmigaNotes by Richard Rae	83
ROOMERS by John Foust	87
Forth! by Jon Bryan Jon completes the graphics portion of the 3D Bouncing Ball	89
68000 Macros on the Amiga by Gerald Hull	97
AMICUS John returns from the WCCA with new tales	103
TDI Modla-2 Amiga Compiler reviewed by Steve Faiwischewski	109

From the editor:

Of Late deliveries:

I received a call the other day (actually I received several of these) and was asked if PiM Publications Inc. was out of business or had we stopped publishing *Amazing Computing*™. It seems that the last issue the reader had received was August.

This came as a surprise to us, we never published an "August" *Amazing Computing*. We had a "printed in July" issue that was received from our printer on August 8th, (in all fairness, the "June" issue was received July 3rd). Our Volume 1 Number 7 issue was shipped in mid September. Number 7 must have been the "August" issue the reader had received.

To further confuse matters, the delivery from our printer of our Volume 1 number 8 was almost two weeks late and was received October 31.

Now, if you are completely confused, you will understand why we went from monthly and numbered designations to numbered issues entirely. I spent a great deal of editorial time (well, they tell me that is my job) explaining the change to worried readers.

If you have been a reader of *Amazing Computing*™ since our "early" days, you have witnessed the slow transformation of our magazine. We made a commitment in those days and we are intent on maintaining it. We will continue to improve the magazine and consistently offer good information. If it requires five weeks to deliver a great product, or four to produce a poor one, we will take the time and do our best.

The first question which is probably springing to mind is, "What about our year's subscription."

Well, PiM Publications has a great many subscribers for such a small start, however, none of these valuable readers has ever received a year's subscription. All subscriptions are sold on a twelve issue basis. If it takes longer than a year to produce 12 issues, they will still receive their issues.

As always, we are working quickly to attain a normal schedule, but it is a great deal of effort so please remain patient.

All of our readers, once told of our problems and solutions, have been extremely complimentary about the magazine. Their answer has been that they were more afraid of our giving up the business, than in not producing on time. They want quality and that is our aim.

Does this mean that there will be fewer typos and misspellings? Maybe.

We are using computerized spelling checkers and friends and associates as proof readers, but there are still mistakes that pass us. Most of the errors are found by us the day we receive the magazines from the printer and not while we are going through the printer's proof.

Our quality is based on the hope that we can maintain a decent magazine with appropriate information and techniques to make the Amiga viable.

We have grown each issue with the abilities of Amiga user's who have written items for us. Each writer has discovered an aspect in the Amiga and a talent in himself, that has made the entire Amiga community a little better.

We have a standing plea, "If you have discovered an aspect of the Amiga that is important and worth knowing, write us. We need you."

Our issue number 9 is jammed with as much information as we could muster. We have included the *Amazing Directory* for our regular readers. We hope the information is helpful.

The *Amazing Directory* is also being sent to dealers around the country in hopes that the extra information on Amiga products may swing a few people to our amazing computer, the Amiga.

At the time I am writing this, we have been promised that this edition will be ready for shipments to arrive at Thanksgiving. Please bear with us, we are trusting the date is true.


Don Hicks
Managing Editor

Software designed for AMIGA.

Lattice® C Compiler **\$225.00**

New version 3.1 of the AMIGA DOS C Compiler replaces version 3.03. Major enhancements include the addition of: TMU, an assembler, a faster linker and version 3 MS-DOS.

With more than 30,000 users worldwide, Lattice C Compilers set the industry standard for MS-DOS software development. Lattice C gives you all you need for development of programs on the AMIGA. Lattice C is a full implementation of Kernighan and Ritchie with the ANSI C extensions and many additional features.

Professional Lattice® C Compiler **\$375.00**

A new product called the Professional Lattice C Compiler is now available. It includes the C Compiler package (complete with TMU), plus LMK, LSE and the Metascope Debugger.

AMIGA C Cross Compiler **\$500.00**

Allows AMIGA development on your MS-DOS system. Price includes the Professional Lattice C Compiler described above.

Lattice Screen Editor (LSE™) **\$100.00**

Designed as a programmer's editor, *Lattice Screen Editor (LSE)* is fast, flexible and easy to learn. *LSE's* multi-window environment provides all the editor functions you need including block moves, pattern searches and "cut and paste." In addition, *LSE* offers special features for programmers such as an error tracking mode and three Assembly Language input modes. You can also create macros or customize keystrokes, menus, and prompts to your style and preferences.

Lattice dBC III™ Library **\$150.00**

The *dBC III library* lets you create, access and update files that are compatible with Ashton-Tate's dBASE system. *dBC III's* C functions let you extend existing dBASE applications or allow your users to process their data using *dBC III* or dBASE III.

Lattice Text Utilities (TMU™) **\$75.00**

Lattice Text Utilities consists of eight software tools to help you manage your text files. GREP searches files for the specified pattern. DIFF compares two files and lists their differences. EXTRACT creates a list of file names to be extracted from the current directory. BUILD creates batch files from a previously generated file name list. WC displays the number of characters and optionally the checksum of a specified file. ED is a line editor which can utilize output from other *TMU* software in an automated batch mode. SPLAT searches files for a specified character string and replaces every occurrence with a specified string. And FILES lists, copies, erases or removes files or entire directory structures which meet the specified conditions.

Lattice Unicalc® Spreadsheet **\$79.95**

Unicalc is a simple-to-operate program that turns your AMIGA computer into an electronic spreadsheet. Using *Unicalc* you can easily create sales reports, expense accounts, balance sheets, or any other reports you had to do manually.

Unicalc offers the versatility you've come to expect from business software, plus the speed and processing power of the AMIGA.

- 8192 row by 256 column processing area
- Comprehensive context-sensitive help screens
- Cells can contain numeric, algebraic formulas and titles
- Foreign language customization for all prompts and messages
- Complete library of algebraic and conditional functions
- Dual window capabilities
- Floating point and scientific notation available
- Complete load, save and print capabilities
- Unique customization capability for your every application
- Full compatibility with other leading spreadsheets
- Full menu and mouse support.

Lattice MacLibrary™ **\$100.00**

The *Lattice MacLibrary™* is a collection of more than sixty C functions which allow you to quickly and efficiently take advantage of the powerful capabilities of the AMIGA.

Even if your knowledge of the AMIGA is limited, *MacLibrary* can ease your job of implementing screens, windows and gadgets by utilizing the functions, examples and sample programs included with the package.

Other *MacLibrary* routines are functionally compatible with the most widely used Apple® Macintosh™ Quickdraw Routines™, Standard File Package and Toolbox Utility Routines enabling you to rapidly convert your Macintosh programs to run on the AMIGA.

Panel™ **\$195.00**

Panel will help you write your screen programs and layer your screen designs with up to ten overlapping images. *Panel's* screen layouts can be assigned to individual windows and may be dynamically loaded from files or compiled into a program. *Panel* will output C source for including in your applications. A monitor and keyboard utility is also included to allow you to customize your applications for other systems.

With Lattice products you get *Lattice Service* including telephone support, notice of new products and enhancements and a 30-day money-back guarantee. Corporate license agreements available.



Lattice

Lattice, Incorporated
Post Office Box 3072
Glen Ellyn, Illinois 60138
(312) 858-7950 TWX 910-291-2190

INTERNATIONAL SALES OFFICES: Benelux: Ines Datacom (32) 2-720-51-61
Japan: Lifeboat, Inc. (03)293-4711 England: Roundhill (0672)54675
France: SFL (1)46-66-11-55 Germany: Pfothenaur (49)7841/5058
Hong Kong: Prima 85258442525 A.I. Soft Korea, Inc. (02)7836372

Lattice

Amazing Mail:

Dear AC:

Hint for Deluxe Paint Users- When having a problem loading a file as a picture, try loading it as a brush. It works for me.

Yours computing,
JJ Shields
Milwaukee, WI

Dear AC:

Our North Seattle Amiga Users' Group has received a letter from a school district asking for help in utilizing the Amiga and in acquiring public domain software for it. Part of our response to the letter is now to ask you if you will please consider inserting somewhere in Amazing Computing™ a blurb such as the following:

The North Seattle Amiga Users' Group passes on an appeal for public domain software and other aid in developing an Amiga curriculum. Send materials to:

Elwood Community School Corporation
ATTN: Amiga Co-ordinator
Rural Route 4, Box 105
Elwood, Indiana 46036
Phone: (317) 552-9861

The above blurb is merely a suggestion, and I am sure that the school district would appreciate the help of Amazing Computing™ in whatever form you deem fit.

This occasion is a chance to make some inroads against the Apple monopoly in the schools. Here in Seattle we intend to mount a vigorous campaign to help this one little school district and then keep track of what develops, for distribution to any other school districts that inquire. Thank you in advance for the help of Amazing Computing™.

Sincerely,
Arthur T. Murray
North Seattle Amiga Users' Group
17050 Second Avenue NW
Seattle, WA 98177

Our pleasure!

Dear Amazing Computing,

I was recently reading the article 'Linking C with Assembly' from your magazine, and started banging my head against the wall around the bottom of page 85. (Volume 1 Number 7) The author mentioned in passing that he felt the AMIGA convention of having the user save registers D0, D1, A0 and A1 (if there values need to be preserved) before calling a library routine, was silly. This was the Zillionth anniversary of all times I've heard or read this complaint. If people just stop and think about it, the convention used by Amiga makes a lot of sense.

The author of the aforementioned article felt that if a called subroutine needs to use these registers, this subroutine should save these registers itself. In this way the push and pop operations would only take place for those registers which are actually modified by the called subroutine. In addition, since the push and pop statements would be encapsulated in the subroutine, the calling program would not need to specify push and pop statements around every call to the subroutine. This leads to smaller and more readable code.

On the other hand I have also heard arguments that no routine should have to preserve registers. In this way the calling procedure would only have to save those registers whose contents had to remain intact. This has the additional advantage that if two or more routines were called one after the other, the important registers would only have to be pushed before the start of the first call, and restored after the last call. This can lead to fewer pushes and pops and therefore quicker code.

I personally feel that the method used by the Amiga software designers is a good compromise between the two extremes. Routines can be

optimized so that data which is temporary in nature can be kept in A0, A1, D0, or D1. Of course the programmer must be on his toes and remember that once a subroutine call to the library is made, these registers can no longer be considered valid. If the user really needs to preserve the data in these registers he can always push their contents on to the stack, and pop them back off when done. But this is rarely necessary because data that is more permanent in nature can be kept in the other registers and will be preserved across the calling action.

Sincerely,
Robert Patterson
Calgary, Alberta
Canada

Dear Amazing Computing™:

A little of that input you covet:

1) If you're going to charge \$4.00 for back issues, it would be nice to know what's in them. People who know the contents aren't going to buy the back issues because they've already read them, and those who don't know what to expect aren't going to part with their money for the mystery behind door number three. Fortunately, this is a Catch-22 with an escape clause. You've already shown the ability to list the contents of disks. Why not do the same for past articles? A simple one line summary of each column or article would help immensely. Later, when you have several volumes behind you, this won't be possible. But for now, give us a break, and give yourself additional sales.

2) As long as you've resorted to the lowly subscription form, make room on it for ordering disks and back issues.

3) The Amiga is an amazing computer, for it can accomodate the neophyte as well as the hacker. For the neophyte's sake, how about a glossary of terms (just once or twice)? I must confess an ignorance of some acronyms used in computerese, and I have read a lot of computer magazines in the last couple of years. Somehow they all take it for granted their readership consists of MIT alumni. Not true. Still, we aren't stupid, just a little ignorant. Some definitions and/or documentation would clear that up right away. (What you need is a professional amateur to screen out jibberish, or at least to point out when some explanation is necessary. I offer my services.)

4) Don't get me wrong. Yours is still the best computer publication I've read, Amiga-specific or otherwise. Hype is kept to a minimum and information seems to be your priority. Kudos to your technical-types: They seem to be able to communicate in proper English, a rarity of the first order. The other possibility, of course, is that they are driving the editors up a wall, in which case kudos go to them instead.

5) Please continue the articles on MIDI. It's still all a mystery to me, and I more or less live in music stores. It would be nice to see in print exactly which hardware configurations do what, and which software is best for specific applications.

Thanks for a great magazine.

Sincerely,
Kenneth E Mitchell
Wahoo, NE

OK, in order:

1) Check out the AC advertisement in the Amazing Directory.

2) Ok, you're right. We placed that in the same Ad.

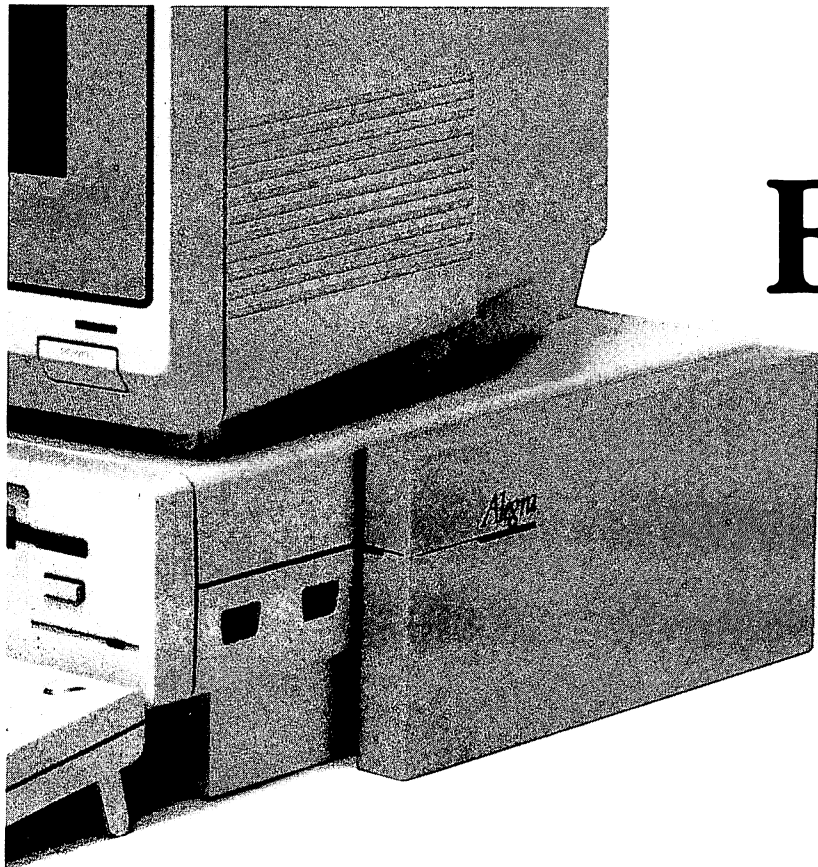
3) We will work out a system to accomodate the beginner, but we ran Amiga terms in number one, the CLI commands, and we do not want to follow other publications who seemed lost in an endless loop of beginner articles.

4) Thanks for the compliment, but our strength is with the expertise of our readers.

5) Richard Rae, our Music Editor, is a fanatic on MIDI (so is our Hardware Editor, Ernest Viveiros), I am afraid you will not be able to escape more MIDI coverage.

Thanks for your insights and input!

•AC•



Expansion Memory Without The Wait.

Introducing *Alegra*: The Amiga™ Memory Expansion Unit from Access Associates.

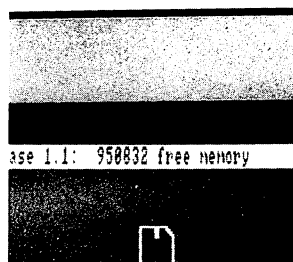
512 K now.

Now you can add 512 K bytes of external memory to your Amiga. In the smallest package available, a footprint only $\frac{3}{4}$ "-wide. And Alegra's no-wait-state design lets your Amiga operate at its intended speed. No delays. With Alegra you get the benefit of fast memory at a surprisingly economical price. AND, BEST OF ALL, IT'S AVAILABLE NOW.

Upgradeable to 2 MB later.

If you'll need 2 MB of memory in the future, Alegra is still the right choice now. Our 2 megabyte upgrade (using 1 megabit DRAMs) will give you the memory you need in the same compact package.

Ask for Alegra at your quality Amiga dealer.



Total system memory is approximately 1 meg with the addition of our 512 K Alegra (depending on specific hardware configurations).

| ACCESS ASSOCIATES
.....

491 Aldo Avenue
Santa Clara, CA 95054-2303
408-727-8520

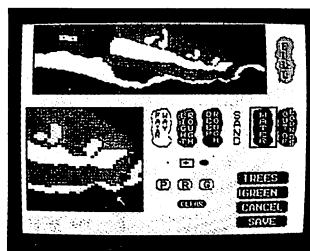
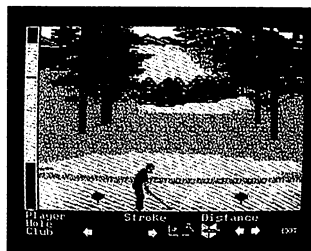
Now, nothing can keep you from your appointed rounds.



Laugh at the weather. Putt in the middle of the night. Tell your secretary to hold all calls. With Accolade's MEAN 18, all the excitement and challenge of real golf is right on your computer.

MEAN 18 delivers the kind of realism and playability you've come to expect from Accolade. This is golf the way it was meant to be enjoyed...

without spending your day decoding the instruction manual. You can hit a bucket of balls at the driving range, play from the pro or regulation tees, even ask your caddy to suggest your clubs.



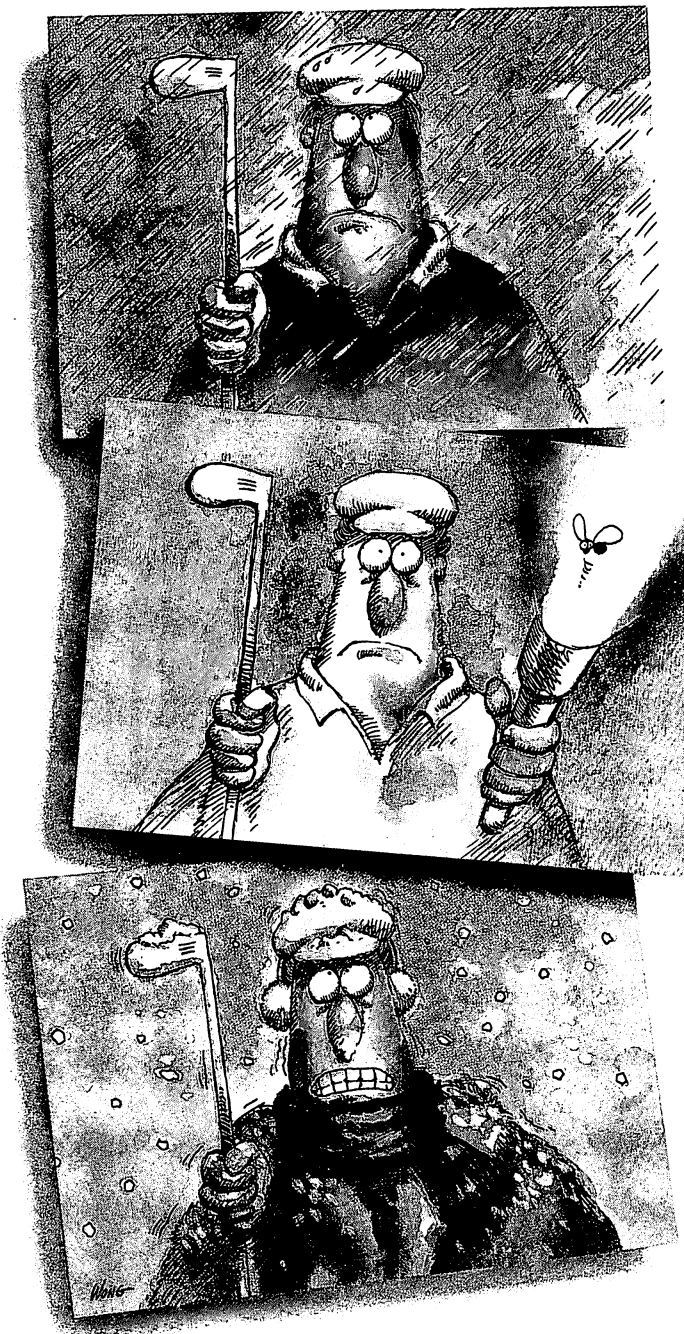
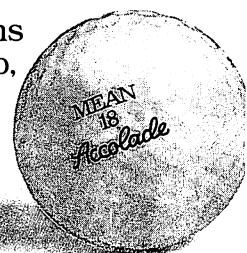
Once you've mastered MEAN 18, there's the challenge of playing on three of the world's legendary golf courses, all capable of bringing any touring pro to

his knees. With The Course Architect, you can even design your own grueling course complete with menacing bunkers and greens on the edge of an ocean.

Available for IBM, Atari ST and Amiga systems.

Accolade, 20833 Stevens Creek Boulevard, Cupertino, California 95014.
Telephone 408-446-5757.

Mean 18.



Amazing Previews...

Making Cents in Business with your AmigaTM

***A quick collection of some of the tools available to
the small business and home Amiga user***

by Ernest P. Viveiros

2+2 Home Management System

The 2+2TM Home Financial Management system (HFM) is a thorough, easy-to-use, integrated home financial package. Using this system, one can track both income and expenses (up to 12 checking accounts) for one full year on one diskette. The program also allows direct checkwriting (checkprinting). Helping establish and maintain a budget, managing credit card purchases and check balancing are among the many uses of HFM. Keeping up to date with your records will allow you to easily retrieve all the necessary summaries for tax time. For piece of mind, included is password privacy protection of your financial data.

HFM is a user-friendly system designed for both first-time users and the semi-professional. Although the program makes no use of the Intuition interface, the program is "easy to swallow" due to the use of the many menus combined with the overall easy-to-follow screen formats. Data entry is easy, and also is correction of such. One of the handiest features of the whole HFM system is the on-line help. Any time help is needed, simply press a function key. This help includes a step-by-step tutorial and reference guide. This feature combined with the comprehensive non-technical documentation, makes HFM a cinch to use.

Special Features

HFM also includes Mailing List Processing, a Personal Calendar and a Telephone Directory.

The mailing list routines allow the maintenance of personal and business mailing lists. The database is pre-formatted to help store a myriad of information including special category coding. The search/selection of records to meet certain criteria is no problem at all. As with any good database, the printing on envelopes or labels is fully supported.

The personal calendar helps to manage and keep track of your time. You may schedule one-time or repetitive appointments. It also warns of potential schedule conflicts. The printing of a daily or monthly schedule is easily done. The calendar also keeps a running record of your activities and allows you to summarize previous and future activities (both total time or time spent with a specific project)

The telephone directory allows the maintenance of all of your important telephone numbers. It is essentially a database in

itself, allowing the search/selection of specific numbers. And of course, the printing of any information needed is supported.

2+2 Home Management System

Olamic Systems Corporation
141 West Jackson Blvd.
Chicago, IL 60604

Impact from Aegis

Aegis has done it again with Aegis Impact, a graphical data management package. Impact is an easy-to-use (yes, even for beginners) package using easy to follow prompts, pull-down menus, and a FAST menu (Of course it makes full use of the intuition interface). Being flexible, Impact allows user input via keyboard, mouse, or a digitizer. This program also will take advantage of extra FAST memory, Genlock (whenever we see that...) and a hard disk drive.

Impact is described as a "graphical data management package". That's a pretty big name. Well Impact does a pretty big job, and a good one at that. But what exactly is Impact and what does it do. Let's take a look.

Impact is basically made up of five parts:

Graph Bulider Table Builder
Icon Builder Slide Builder
Slideshow

The Graph Builder allows you to create a variety of charts (Bar, Line, Area, Scattergrams, and Pie Charts) These charts can be overlayed, stacked or even displayed in 3D. You can choose colors, patterns, and axis polarity, axis step rates, graph size and much more.

The Table Builder includes a text editor which allows you to make the labeling easy as pie. Options include multiple fonts, sizes, and styles. Also text formatting and editing is supported.

The Icon Builder allows you to create icons to use as s Brush or stamp in your works.

The Slide Builder and Slideshow are the "meat and potatoes" of this package. The slide builder allows you combine charts,

tables, and icons with many drawing tools such as lines and circles. It's basically a fine tuning tool for the individual graphs.

Next and finally comes the sildeshow. This allows you to put all of your individual graphs into a "slideshow". Many types of effects are available for the slideshow including "wipes" , "fades" and "spirals".

This will really give them a presentation that will knock their socks off. Remember a super graphics presentation is worth a thousand raises.....

Aegis Impact \$199.95
Aegis Development
2210 Wilshire Blvd.
(213) 306 0735

The Rags to Riches Accounting Series:

Ledger Module

Payables Module

Receivables Module

For a "full fledged" accounting system, the Rags to Riches Accounting Series consists of three modules, each of which can be used independently or may work with the others to become an integrated accounting system. Chang labs also offers this system in a Macintosh version.

The Ledger module helps to keep all of your transactions and balances in order. It is very easy to use. Each account entered is a key, not a number that you will have to remember. And the system is so easy to use that if you need some information, just point to what you want (using the keyboard, not the mouse), and press another key. That's all there is to it.

The Payables module will allow you to keep each vendor up to date as you work. This way you can keep track of your cash flow, and also keep away those nasty phone calls and letters. Once again, each vendor is identified with a key, and to see which bills are due/paid, just point and press a key. Once you have paid the bills, merge the results into the ledger module.

The Receivables module gives you instant updates of your customer balances and it will also print reports of such. Again each is identified with a key and information is dug out with a point and press.

Rags to Riches is easy and comfortable to use. The best part is, it is expandable.

The Rags to Riches Accounting Series:

Ledger Module

Payables Module

Receivables Module

Chang Labs
5300 Stevens Creek Blvd
San Jose, CA 95129
(408) 246-8020

Phasar (Programmed Home Accounting System and Register)

PHASER is another easy-to-use home finance manager of your accounts. It can also determine a budget, calculate and project your taxes, analyze finances and keep your affairs organized.

Using PHASER is just like using a checkbook with the exeception of having the calculations to worry about. The transactions are enter into columns which is similar to a check register.

PHASER also has budgeting capabilities which are designed to help you achieve your financial goals. Income/Expense/ Budgeting categories can be easily edited at any time. There is also a feature for analyzing cash transactions. PHASAR also has a full range of reports and graphs.

PHASAR can give you a list of all transactions for any combination of budget categories, accounts, payees or date range to help you with your taxes. It can also do a tax calculation based on the entered transactions (up to ten different tax calculations). PHASAR also allows you to estimate your year-end tax standing at any time during the year.

PHASAR also includes a couple of handy features: Loan and Savings Analysis/Net Worth Presentation/Calandar/Phone Book

PHASER in summary is a complete personal finance manager which is fast, easy-to-use.

Phasar (Programmed Home Accounting System and Register)
Financial Manager
List \$89.95
Requires 512k and one disk drive

Marksman Technology, Inc.
Route 5, Box 221A
Santa Fe, NM 87501
(505) 455-2681

Money Mentor

"Keep track of your pennies and your dollars will take care of themselves!". This old (but good) advise, is the premise upon which Money Mentor, a personal finance manager, is based. Using the full capabilities of the Amiga, Money Mentor will analyze and graph your financial situation.

If you are one of the many who shy away from personal finance packages because of the tedious data entry that is usually required, then this may be the package for you. Money Mentor has a unique system called "Smart Scrolls" which studies previous transactions and remembers details about them. This can help save up to 70% of the typing typically required.

The heart of Money Mentor is in three main "systems": Budget, Transaction, and Reporting.

The Budget system allows the use of up to 200 budget categories (100 income/100 expense) for any fiscal year. These can be created, edited or deleted at any time.

The Transaction system allows the management of up to 30 accounts (checking, cash, savings, credit cards, etc...). Transfers between accounts are supported. Also supported is a search routine which allows the editing or totaling of accounts to specific conditions. Another included feature is check printing. The system also features automatic Account Balancing.

The real beauty of Money Mentor is in the Reporting System. This system lets you see your financial situation through many different colorful graphic reports. These can be used to understand and project your finances. Money Mentor also allows the printing of over 50 different reports, for those who are information hungry. Money Mentor really can "help you keep track of your pennies."

Money Mentor List \$95.95
Sedona Software
11844 Rancho Bernardo Road #20
San Diego, CA 92128-9901

Database Deluge

It seems that every software company has a database package. Why? Well because almost anyone could effectively use a database. Doesn't it make sense to have well organized files that can be searched and manipulated quickly and easily. Of course it does! That's why many people use databases, but also many do not when they should. Why?

One reason might be that people do not know which database to buy. There are so many databases on the market, and so many new ones entering every day that it is mass confusion. So you say to yourself, "If you've seen one database, then you've seen them all". Let's dispell a few misconceptions.

Most databases have the same basic features: editing of entries, search, print records, delete records, etc... Yet all databases are not the same. Different databases have different specializations (math functions/graphics/mail merge/Etc...) It is these specializations along with the ease-of-use, that you should shop for in a database. There are many to choose from out there... think of what you need it for... then go searching...

Here's a quick look at the special features of a few of the databases on the market today:

Organize!

This database is an easy-to-use information manager, featuring powerful math functions. Organize! uses the Intuition interface (pull-down menus) and is easy-to-follow. Also supported are customized forms and reports. Organize! also includes password protection.

Organize! is based on the same premises as other Micro Systems Software's other Amiga products. This makes it highly compatible with Analyze! (spreadsheet), Online! (communications), and Scribble! (word processor).

Organize! list \$99.95
Micro-Systems Software, Inc.
4301-18 Oak Circle
Boca Raton, FL 33431
(800) 327-8724

AMIGA LaserPrinting!

Do this! From your Amiga™ to an Apple LaserWriter™ or a LaserWriter Plus™. Plain or fancy, near-typeset quality printing can be done easily from your Amiga word processor or text editor with a few commands embedded in your text using LaserUtilities Vol. 1.

Some of the features of LaserUtilities Vol. 1 include:

- Full justified, centered, or ragged-right lines or paragraphs.
- Works with Ed, TxE, EMACS, and most other editors.
- Within-line or within-paragraph font-changes.
- Automatic page break and numbering if desired.
- Filled or open boxes and bullets of any size.
- Lines and other PostScript™ graphics.
- Boxed-in lines or paragraphs.
- Easy font selection and unlimited font scaling procedures.
- All procedures extensible and editable.
- Bonus useful PostScript programs on each disk.

Also available are LaserFonts Vol. 1 (\$39.95), a disk of three downloadable decorative LaserWriter fonts, and LaserUp! Graphics (\$79.95), a complete Amiga to LaserWriter graphics screen printing package.

For more information or to place an order please write or call:

S. Anthony Studios
Scott Anthony
889 De Haro St., San Francisco, CA 94107
(415) 826-6193

LaserUtilities Vol. 1 - \$39.95 retail

All orders, please add sales tax where applicable and \$2.50 shipping per disk. Dealer prices on request.

LaserWriter and PostScript software also available for Apple II™ and MS-DOS.

OmegaFile

OmegaFile is a database with mail merge capabilities. This program does not use the Intuition interface, but is still user friendly. The power in this package is in the sorting and in the mail merge capabilities (they really are powerful). Also featured are powerful mathematical functions.

OmegaFile

The Other Guy's
P.O. Box H
Logan, UT 84321
(800) 942-9401

Info+

This is your basic database, but a very good one at that. It makes use of the intuition interface, and is as user-friendly as can be. There is really nothing special about this program except that it cuts through all frills, and lets you maintain a database without all the lights and hype. It's fast, quick (but, by no means is it dirty), and accurate. It is definately a good buy. Check it out!

Info+ \$49.99
Eastern Telecom
9514 Brimton Drive
Orlando, FL 32817
(305) 657-4355

•AC•

Alien Fires

On December 1st
the most sophisticated
adventure game series ever
made will be unleashed for
your Amiga.

We invite you to experience
the first in this series, the
most captivating and realistic
role-playing adventure ever
dreamed of.

*Alien Fires —
Part I, 2199 AD*

By JAGWARE

Look for our full-feature ad
in the December issue or, if
you just can't wait, write or
call us for more information.

JAGWARE INC.

288-2 Montreal Rd.
Ottawa, Canada
K1L 6B9
(613) 744-7746



Dos 2 Dos

Reads and Writes IBM Disks

*"...does what you thought Transformer would do...
transfer files between PC/MS-DOS and AmigaDOS."*

By Richard Knepper

As a computer purist, I opted to purchase an Amazing Amiga rather than making the "rational" choice of an IBM compatible. Unfortunately, much of my work requires IBM compatibility. The Amiga Transformer met those needs by giving me a limited amount of compatibility, at the expense of speed and graphics.

But I soon found that this was not enough. The programs written on the Amiga are superior to those on the IBM, and working on a slow version of a poor program was no fun at all. The Sidecar promises a return to normal IBM speeds, but there is still the problem of having to use those inferior IBM programs.

Coming to the rescue is a company called **Central Coast Software**, which produces a program which they promise "does what you thought Transformer would do... transfer files between PC/MS-DOS and AmigaDOS." The program they have written is called **Dos 2 Dos**, and, with some qualifications, meets their claims.

Dos 2 Dos is a utility program that allows the Amiga user to transfer files between MS-DOS and AmigaDOS disks. The program requires 256K memory and at least two disk drives.

Dos 2 Dos is a CLI-only program, meaning that there is no Intuition interface, no pull-down windows, and the mouse isn't used. Dos 2 Dos doesn't translate the files, so only ASCII and binary files can be transferred. One can't transfer an IBM program to the Amiga, and then run it.

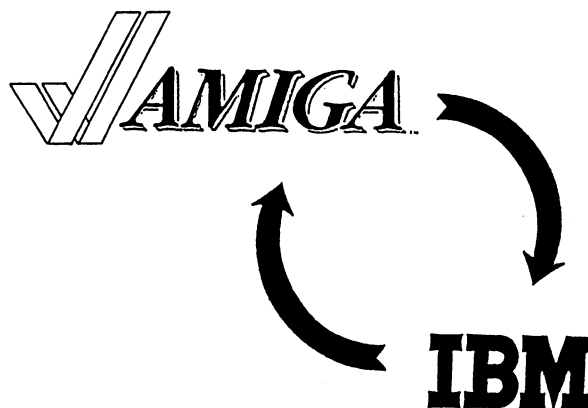
The program rewrites the disk drive controllers, so multitasking with Dos 2 Dos may well send you on a quick trip to the Guru. Also, the drives aren't reprogrammed when the program is ended, so the computer has to be rebooted in order to use the drive you read from. Dos 2 Dos supports both 5 1/4 and 3 1/2 inch disk drives. The 3 1/2 format supports both the 720K and 360K formats. Both formats are available under the Amiga Transformer.

The manual suggests that Dos 2 Dos be copied to the Workbench disk. This is a very good idea, especially for users with only one disk drive. For those with one disk drive and no working knowledge of AmigaDOS, use the following procedure:

Open CLI from the Workbench disk.

Type the following commands:

```
Makedir ram:c
Copy c ram:c
Path add ram:c
```



Insert the Dos 2 Dos disk and type:

```
Cd df0:
Copy df0:d2d ram:
```

Put the Workbench disk back in and type:

```
copy ram:d2d df0:
```

Dos 2 Dos can then be loaded by opening CLI from Workbench and entering 'd2d'. It can also be opened by breaking out of the startup-sequence, by hitting CTRL-D and then entering 'd2d'.

Dos 2 Dos uses both AmigaDOS and MS-DOS command structures for copying files. For those unfamiliar with these commands, a short synopsis and critique of each will be given.

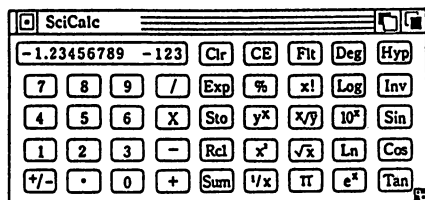
CD

Changes the current directory of the Amiga and MS-DOS disks. The path structure for each is akin to their native operating system. For example, to change the Amiga directory from `df0:` to the subdirectory `df0:foo/foo2/`, one would type:

```
CD df0:foo/foo2/
```

SciCalc™

Scientific Calculator For The Amiga™



\$19.95

Don't let the price fool you! SciCalc has full algebraic hierarchy and features an automatic constant that is a delight to use. Choose from 3 display modes: Floating Point, Scientific, or Fixed Point..

Press the Hyp(erbolic) key twice and a whole new page of functions is at your fingertips. No long waits - SciCalc has been available since March. Your order with manual will be sent by First Class mail.

Features

- Large Equals Key (Display)
- Adjustable Size
- 10 Memories
- Powers
- Logarithms
- Trigonometry (D/R/G)
- Hyperbolics
- Polar/Rectangular Conversions, and more.
- Color Highlighting
- Full Error Trapping
- 2 Dimensional Statistics
- Linear Regression
- Linear Estimation
- Correlation Coefficient
- Factorials to 170

Dealers Inquires Welcome

Send Check for \$19.95 to:

DESKWARE

P.O. Box 47577

St. Petersburg, FL. 33743

The copy command supports both MS-DOS and AmigaDOS wildcards. For MS-DOS they are:

? - any number of characters up to the limit of the filename or the filetype

***** - specifies any file name or file type for AmigaDOS:

? - any single character

#? - any number of characters

Unfortunately, Dos 2 Dos does not copy MS-DOS subdirectories. Using **Copy df1:*. *** will copy only the files in the current directory. Therefore, it is necessary to 'CD' into each directory path of the disk to copy all of the files.

There are two subcommands which are especially useful. The first is the **-A** command which strips the control characters, excepting tabs and line feeds. This translation will also convert Wordstar files into a format readily usable by most Amiga word processors.

Secondly, **-R** will automatically replace files on the destination disk. This is especially useful when one just copied a bunch of Wordstar files and forgot to use the **-A** command.

DIR

Displays the contents of the directory stated. It displays them in MS-DOS format, so dates and disk space free are also displayed. The command supports paths in the same manner as the 'CD' command.

DELETE

This command deletes a file. Unfortunately, wildcards are not supported, so deleting files using this program can be an arduous task.

EXIT (or X)

This exits Dos 2 Dos and returns you to the CLI. Since your disk drive will be all screwed up, you will have to reboot your computer to regain normal use of the drive that previously held the IBM format disk.

FORMAT

This formats MS-DOS disks. There are two useful options. The first is **/1**, which specifies that the disk is to be single sided. The second is **/8**, which specifies that the disk should have 8 sectors per track instead of nine. These options are standard MS-DOS FORMAT options.

HELP - List all Dos 2 Dos commands and their syntax.

TYPE - Displays the contents of a text file on the screen. It may be used to display either AmigaDOS or MS-DOS files. Paths can be specified, but wildcards aren't accepted. While other files can be displayed, only ASCII files will be legible, of course.

To change the MS-DOS directory from **df1:** to the subdirectory **df1:foo/foo2**, one has to type:

```
CD df1:\foo\foo2
```

This is standard MS-DOS command format for changing directories.

COPY

Copies files from MS-DOS to AmigaDOS, and vice versa. The command structure is the same as for changing directories, and this can lead to quite a bit of confusion. For example, to copy the MS-DOS file **foo**. you to AmigaDOS, you might have to type:

```
Copy df1:\foo\foo2\foo.you df0:foo/foo2
```

It is very easy to get the normal slash mixed up with the backslash and screw everything up. Fortunately, the program supports default directories. This means that the program will copy the file to the default directory if none is stated. This is especially good because the program sometimes does not listen to commands and does not copy into sub-directories, but instead copies to the main directory.

Dos 2 Dos has a number of limitations. The most important is that only one drive may be designated as the MS-DOS drive. This means that the program will not transfer MS-DOS to MS-DOS. It also defaults the AmigaDOS drive to **d:f0**: No external drives may be used as the destination. Dos 2 Dos doesn't support quad-density disks, meaning IBM-AT files may not be transferred directly.

MS-DOS disks that use non-standard sector sizes or tracks beyond forty may not be used. Finally, files cannot be copied from or to the ram: disk. Although none of these limitations are of any great consequence, it would be nice if future revisions of the program would overcome these problems.

I have a some major complaints about Dos 2 Dos. The first and foremost is that it does not take advantage of the Intuition interface. It would have been easy to create an overlay program that would allow paths and program names to be specified with the mouse. We've all seen these directory utilities being used in other programs. There is simply no excuse for an Amiga program having such a poor user interface.

The lack of multitasking can be circumvented. It is possible to run other tasks concurrently, but the system may crash. Central Coast Software should take care of this problem, either making multitasking less bug-ridden, or else preventing it altogether.

Dos 2 Dos should have a partial Workbench. Since it doesn't multitask well, the program could be located in the C subdirectory and booted during the startup-sequence. This would work better since one needs to reboot Workbench after using the program.

All problems aside, Dos 2 Dos does the job it is intended to do. I have successfully transferred Lotus files to Maxiplan and Wordstar files to both MicroEmacs and Scribble!. Textcraft is a problem because it has trouble reading ASCII files. The way around this is by loading Textcraft from the CLI and entering

```
1> textcraft filename filetype=ASCII
```

Once in Textcraft, the program can be saved and used normally. It should be noted that this problem is not the fault of Dos 2 Dos, but rather a general one encountered whenever you port files around like this.

Dos 2 Dos does seem a bit quirky in that the destination AmigaDOS disk may acquire a read/write error. This problem can be circumvented by recopying the files to a second disk using AmigaDOS. I have had no problems with the files once this is done. However, this read/write error happened twice, on separate disks, while I was testing the program.

The success or failure of a utility program rests solely on whether or not it works. Since Dos 2 Dos does work, I would recommend it. Another reason to give it the thumbs up is that the product is currently available, not just a promise of things to come. People who use MS-DOS at work and an Amiga at home will find this program extremely useful.

Even the Masters would be Amazed.



EASYL™

PRESSURE SENSITIVE DRAWING PAD FOR THE AMIGA™ PERSONAL COMPUTER.

FOR USE BY AD AND FILM DESIGNERS, ANIMATORS, ARTISTS AND STUDENTS. WORKS WITH ALL AMIGA GRAPHIC AND CAD SOFTWARE. EASYL™ BY ANAKIN RESEARCH, INC., 100 WESTMORE DR., UNIT 11C, REXDALE, ONTARIO, CANADA, M9V 5C3, (416) 744-4246
— DEALER INQUIRIES WELCOME.

New Amiga Books

NOW AVAILABLE!
The Amiga System

The Amiga System: An Introduction by Bill Donald is now available from Progressive Peripherals & Software, Inc.!

This book is a storehouse of technical information about the Amiga computer and its operating system. If you have been looking for in depth information on the newest 32 bit computer available today, the **Amiga System: An Introduction** is the best source of information you could obtain.

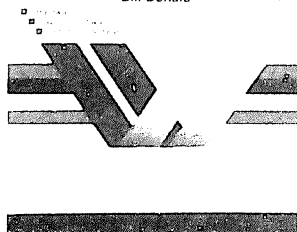
NOW \$15.95

Suggested Retail Price

THE AMIGA SYSTEM

An Introduction

Bill Donald



NEW!

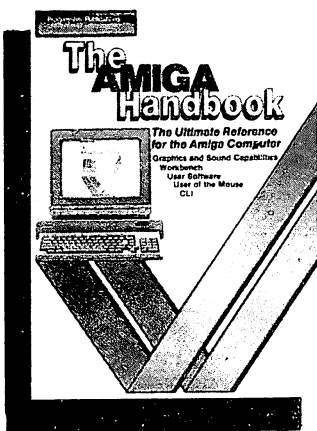
THE Amiga HANDBOOK!

NOW AVAILABLE!

\$24.95

Suggested Retail Price

The Amiga Handbook contains all the information you need to get the most out of your Amiga. It is a well thought out and clearly written book to give you the information they never included in the Amiga documentation. This book provides a complete, detailed reference source of the Amiga and its operating system. If you own an Amiga, or are considering purchasing one, this book is a must!



The Amiga Handbook Includes:

Description of the System Architecture—Amiga Workbench Discussion—Intuition: Basis of the Amiga—The Graphics Programs Graficraft and Deluxe Paint—Amiga for the Advanced User—The Graphics User Interface—Understanding the CLI—Automation of the Amiga (Command Sequences)—The Special Chips of Amiga: Denise, Paula and Agnes—Basics of Sound and Graphics—Programming the Amiga (Amiga Basic from Mirosoft, Lattice)

The Amiga Handbook contains detailed descriptions of the Amiga Systems. This will help you with your purchasing decision and appraisal of this new computer. For others, it is written as a handbook that contains many tables and exhibits which will help you work with the Amiga on a daily basis. It is a valuable aid that will help you learn and work with the Amiga quickly and unhindered.

This book contains well over 400 pages of information to aid you.

It is compulsory reading for everyone who has an interest in the new Amiga Super Computer.



Call Today
(303)825-4144

I feel that this program circumvents my need to buy Sidecar or use Transformer. Based upon this, one could conclude that Dos 2 Dos is worth more than the \$55.00 retail price. I do not. The program just doesn't have the feel of a professional program.

The thought of using MS-DOS data in Amiga programs is very appealing, especially to those who do not like the quality of MS-DOS programs. Hopefully Central Coast Software will dispel all my doubts about the product with future revisions.

They have the beginnings of a product worth more than the \$55.00 asking price. If they take care of the bugs and develop a better interface, they could end up with a program that would provide a very viable alternative to the Amiga Transformer and Sidecar.

Dos 2 Dos

\$55.00

256K Amiga with 2 disk drives required

Central Coast Software

268 Bowie Drive
Los Osos, CA 93402
(805) 528-4906

•AC•

...editor's note

By John Foust

As this review crossed my desk, so to speak, I got word that the Workbench 1.2 upgrade will be composed of three disks and some documentation. This includes the new Kickstart and Workbench, along with a new Extras disk. This upgrade is expected to sell for \$15.00.

The Extras disk is removed to include a program to read and write IBM disks. At this time, and at last word, this program does not read IBM formatted 3 1/2 disks. It will only work with 5 1/4 disks. Please note that Dos 2 Dos does read 3 1/2 IBM disks.

This Commodore utility will take over the drive much in the same way as the Dos 2 Dos utility. It is known to have less extensive wildcarding abilities than Dos 2 Dos, according to an internal Commodore source.

Please keep in mind that both these products do not perform Transformer-style emulation of programs. You are free to transfer both binary and text files between disks. Since program files are designed for different microprocessors and operating systems, they will not execute on the other system.

In many cases, with the text file translation abilities of both products, word processing files can be moved between disks of otherwise incompatible operating systems.

•AC•

MaxiPlan

"...very similar to Lotus 1-2-3 in execution, but at the same time takes full advantage of the power of the Amiga."

Reviewed by
Richard Knepper

MaxiPlan is a combination worksheet, database, and graphics program, from **Maxisoft** and distributed by **Electronic Arts**. It is very similar to Lotus 1-2-3 in execution, but at the same time takes full advantage of the power of the Amiga. It multitasks, uses pull-down menus, and has color and sound. Multiple spreadsheets can be used at once, and information can be cut, copied, and pasted between them. A separate macro program can also be used to customize the worksheets for specialized uses.

MaxiPlan requires a 512K Amiga, and one disk drive. More memory and two disk drives are recommended to take advantage of MaxiPlan's full potential. A color printer is also useful, but a black and white printer can be used. The program is not copy protected.

MaxiPlan is an enormous worksheet. Most spreadsheets have a million or so cells, which is more than you ever need to use. Maxisoft was not content with a million cells, and created a worksheet that has over 8 million cells available. Fortunately, memory is only allocated to those cells that contain information, and a 512K Amiga will have about 170K free after loading an enhanced worksheet.

The program loads directly from Workbench. It can either be booted after Kickstarting the Amiga, or else loaded from your normal Workbench disk. When MaxiPlan loads it attempts to close Workbench to save on memory. It is generally a good idea to boot MaxiPlan from its own Workbench so that you will have the maximum amount of memory to work with. This is especially true if you plan on using more than one spreadsheet at once.

After MaxiPlan is loaded, you are presented with a control window. From this you can load an existing worksheet, or open a new one. You are also given a choice of using four or eight colors in your worksheet. MaxiPlan defaults to four colors in enhanced mode, but this requires more memory.

The Worksheet

The MaxiPlan worksheet is very similar to Lotus in appearance. For those unfamiliar with spreadsheets, there are a number of columns across the top and a number of rows along the side. The column and row coordinates serve to delineate cells, into which data, formula, and labels may be entered.

The MaxiPlan worksheet has a number of differences that serves to make it unique. It has both vertical and horizontal scroll bars. There is a sizing gadget, which is very useful when more than one

worksheet or graph is being displayed. Page forward, page back, and close window gadgets are also present.

Across the top is the familiar Amiga menu bar. Just below the menu bar is an area that displays the currently selected cell and its contents. This area has a number of formula entry buttons which allow entry of formulas using the mouse alone. This is very handy. The buttons alone allow you to access the built-in functions of the program.

The MaxiPlan worksheet is based on the concept of ranges. All commands will either affect a single cell or a range of cells. There are no global commands per se. To accomplish global formatting, all cells of the worksheet must be selected. Only then can an attribute be specified, such as width. Although this will at first be an annoyance to those use to standard spreadsheets, the idea of ranges becomes very appealing after a bit of practice.

The worksheet also has a very handy feature called zoom. Pressing the zoom button will allow you to view a pictorial representation of the worksheet. Each cell becomes one character wide. Those cells that contain information are highlighted with different colors to show labels, formulas, and data. This feature is useful not only for formatting your spreadsheet, but also for moving about. Just click on any cell and then click the normal button, and the worksheet will put the selected cell in the upper left-hand corner.

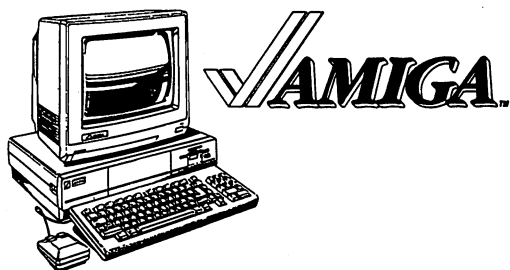
There are a multitude of ways that you can move around the worksheet. The first and most obvious is by using the arrow keys. The mouse may also be used to click on the desired cell. Wordstar keys may be enabled to allow the use of Control-S, -X, -D, and -E to move left, down, right, and up. Finally, there is a "goto" command that allows the specification of a particular cell.

The use of menus sets MaxiPlan apart from all other spreadsheets, even those whose revisions have included mouse support. MaxiPlan was obviously designed with the mouse in mind. The menus are laid out logically, allowing for easy access of the special features of the program.

The first menu, the Project Menu, allows you to load and save worksheets. It also permits additional worksheets to be opened. Multiple sheets can be opened at one time, but a 512K Amiga is effectively limited to having two active at once. Printing options are also located in this menu. Special print features, such as titles and the printing of grid lines, can be used.

Century Systems

has the best prices and service in the USA on the incredible AMIGA Computer.



Many Amiga's and Amiga expansion products available.

Prices too low to print. And service too good to ignore.

Century Systems
8033 University Ave.
Des Moines, IA 50311

SALES 1-800-223-8088

24 HR. SERVICE 1-515-223-8088

The Edit Menu is very similar to the edit menus appearing in many other programs. It is good to see Maxiplan using the standardized Amiga menu style, and hopefully all Amiga programs will soon incorporate edit menus. Using this menu, you can cut, copy, and paste regions and cells. You can also define names for cells and ranges. These names may then be incorporated into formulas, and ranges may be moved without having to change references to them.

The Format menu controls the manner by which cell contents will be displayed. MaxiPlan has the normal formats, such as general, dollars, fixed, and time. It also allows widths to be designated. Either single cells or ranges can be formatted using this menu.

This menu also contains two commands which set MaxiPlan apart from most other spreadsheets. First, a type style may be selected for all values, labels, and formulas in a particular range. Bold, underline, italics, or any combination of the three may be selected. A second command will allow you to select the color of a particular cell or range. This is especially useful for highlighting important spreadsheet data.

The Options Menu also has a few special commands that allow you to show off the power of Maxiplan and the Amiga. First, the talking command allows cell notes to be spoken, written, or spoken and written - more on notes later. The keyboard echo command allows keyboard input to be echoed verbally. This is very useful when inputting large amounts of data, as it allows you to orally check your accuracy.

The Commands Menu contains most everything not covered in the first four menus. The first important feature is the note command. This allows the attachment of notes to specific cells. When a note is selected and the help key is pressed, the note will appear in the form of a requester. The use of notes is invaluable when creating worksheets for others to use, and is perhaps one of the best attributes of the program. This menu will also allow the specification and drawing of charts.

The Data menu accesses the database functions of Maxiplan. This menu also allows saving of data as text, which can then be incorporated into other programs.

The last menu, the Macros menu, is disabled. This is because macros have not been included in Version 1.0. They are available as a separate program. According to Mike Lehman, a principle designer of MaxiPlan, macros should be included in later revisions of the program.

Maxiplan Graphics

MaxiPlan supports four different graph types. These are Bar, Line, Pie, and Area graphs. Each graph may have a title, as well as a bottom and left label. Four rows of data may be plotted against as many as twenty columns. In version 1.0, no provision is made for X-Y plotting.

A nice feature is that more than one chart may be dynamically linked to a worksheet. Then, whenever worksheet information is changed, the graphs are automatically updated. The combination of multiple charts and dynamic linkage is a real boon for those of us who like to tell lies with statistics.

Charts may be printed to black and white or color printers. It seems a shame to have to use a single color to print such nice looking graphs. Most MaxiPlan power users are going to find that they won't be satisfied with black and white graphs, and will consider the purchase of a color printer.

Perhaps the best feature is the ability to capture any graph as an IFF interleaved bit map file. The file can be exported to paint programs such as Deluxe Paint or Aegis Images. By using Maxiplan to create graphs and then modifying them with a paint program, professional charts can be created.

Database Capabilities

MaxiPlan has a number of built-in database capabilities. There are a number of functions and commands that will allow you to manipulate data. First you must specify a data range. Once this is done you may then input records. The database commands will then allow you to find, extract, delete, and sort the records.

A good application for the database is as a customer file. Customer records could be extracted according to specific sort criteria, and these records could then be used for form letter generation. This is a very powerful feature, and Maxiplan's database provides all the power needed for many applications.

MaxiPlan Extras

The program includes two utility programs that are very useful. The first one is called MaxiMerge. It is used to merge MaxiPlan database data with other text files. A form letter can be created using ED or Textcraft and merged with the customer information extracted from the database. The program will then print letters that incorporate the customer information.

The second utility is called "From 123" and, as you may have guessed, imports data created on Lotus spreadsheets. The program does not read MS-DOS formats, so the file must be transferred to AmigaDOS first. I've used DOS 2 DOS, a program reviewed elsewhere in this issue.

There are also rumors that the final Workbench 1.2 will have a file conversion utility. If all else fails, you can simply download the lotus file using a telecommunications program. The capability of importing data is very important for those of us who must use MS-DOS. Note that the utility imports data, and as of revision 1.1, cannot be used to export data.

Some Problems with MaxiPlan

You may have noticed that until now I have not mentioned the program manual. This is intentional, because I consider the documentation inadequate. MaxiPlan is a very powerful program, and the manual doesn't explain the methods by which this power can be harnessed.

The first section provides a tutorial called "Introduction to MaxiPlan", but not enough examples are provided to allow you to become comfortable with the program. The second section of the manual is the "MaxiPlan Reference Guide", and is a less cutsie description of the program. Again, not enough examples are provided. The authors seem to have forgotten that vicarious learning is the easiest way to master something new. If you are not familiar with the workings of spreadsheets, you can expect to spend quite a bit of time becoming used to the peculiarities of MaxiPlan.

My second problem is that the program may have a few bugs. They existed in Version 1.0, but I think they have been removed from the revision. Some of the bugs are most likely due to the quirkiness of Workbench 1.1, so MaxiPlan cannot be blamed for them. In fact, they have gone far to avoid certain types of system crashes - out of memory errors, for example.

Although the graphs created by the program are very good, the means by which they are specified are too rigid. The data to form the graph must be ordered in rows, and they can only be plotted against twenty columns. Whereas Lotus allows data to be used for delineating the horizontal axis, MaxiPlan only allows labels to be used. This means that you will have to create labels for each graph you make. This takes quite a bit of time, and can be very annoying. There is also no provision for manual scaling, which is very useful for telling statistical lies. There is nothing as useful as expanding the axis to make a crooked line look straight.

MaxiMerge doesn't seem to work correctly in 1.0. There seems to be a problem addressing the two files and the printer simultaneously. In any event, expect a trip to the guru everytime you try this utility. Those with the revision shouldn't have this problem, but users with the original won't be able to use MaxiMerge.

Haven't You Set Your *AMIGA'S* Time And Date Once Too Often?

Introducing

A - T I M E

*A clock/calendar card with battery back-up,
so you will never have to set the time and date
in your AMIGA, EVER AGAIN!*

- Plugs into the parallel port.
- A completely transparent printer port is provided, with total compatibility to all I/O operations.
- Battery back-up keeps the clock/calendar date valid on power down.
- Custom case with a footprint of only 2 1/4" x 7/8" x 3 1/4" (W x D x H) in standard *AMIGA* color.
- Leap year capability.
- A - T I M E package contains:
 - 1-A - T I M E clock/calendar module
 - 1-3.5" DS Utilities Disk
 - Operating instructions

PRICE \$59.95

AVAILABLE: NOW

Mail check to:

AKRON SYSTEMS DEVELOPMENT (ASD)

P. O. BOX 6408

(409) 833-2686

BEAUMONT, TEXAS 77705

include \$3.50 for shipping and handling
For MC/VISA orders call (409) 833-2686
AMIGA is a trademark of Commodore - Amiga inc.

I was upset to see that the MaxiPlan macros were not included. As they are an invaluable tool, every owner will want it. I would rather pay for them up front than have to order them and wait. MaxiPlan would still be a bargain even with the extra cost of the Macros, and they should be included. It is akin to buying a Porsche and finding out later that the engine wasn't included.

Hope For the Future

MaxiPlan seems dedicated to user support. Most of the bugs have been eliminated, and they are currently working on a low-cost upgrade. The upgrade will eliminate the remaining bugs (it will only work with Workbench 1.2, which is a big help in bug-killing,) and the product should eliminate most of the problems that I have with the program.

A note to current MaxiPlan owners. If you have sent in your owner card you should have received the revised MaxiPlan 1.1. If you have not, you should contact Maxisoft and they will make sure you receive it immediately. They have also shown that they have a strong commitment to user support. Should you have any specialized problems that you cannot seem to solve, give them a call and they will do their best to help you solve them.

Upcoming macro ability

Maxisoft is also selling a macros program to be used in MaxiPlan. It is available for \$25.00, and may be purchased directly from Maxisoft. This program adds considerably to the capabilities of the program. It is especially useful for the creation of custom worksheets.

Macros resides as a separate window and can operate upon multiple worksheets. Up to 32 macros may be specified, and a learn mode is provided. The learn mode allows Maxiplan to remember a number of keyboard and mouse commands, and execute them upon activation of the Macro.

The macros that can be created are extremely powerful. They can have subroutine calls, and allow for the creation of loops. Data can be entered and then updated in real-time using loops and formula step rates.

Another feature is that CLI commands can be executed from within the macros. This provides true multitasking capabilities to the user. IFF files may be viewed using the CLI, but they, by necessity, take up the entire screen and must be closed before the macro may proceed. Finally, by starting the macro in the upper left-hand corner, execution is automatic upon loading. This means that when you load a program, it could say "hello" to you, show relevant graphs, and print information to the printer. It almost makes your Amiga sing and dance!

Anyone familiar with Lotus macros will find the purchase of this program a necessity. The power provided by this program is phenomenal. It should also be able to execute Lotus files when spreadsheets are imported (more on this later), although I have not personally tested this.

Conclusion

MaxiPlan and MaxiPlan Macros together only cost \$175.00, but there are two hidden costs involved when purchasing the program. You will probably want to buy a new color printer and expanded memory. You will need the memory to take full advantage of MaxiPlan's capabilities. MaxiPlan has been tested with the Alegra expansion board, and Maxisoft claims that it works with Tecmar and Comspec boards. Although these are by no means a necessity, MaxiPlan gives you a good excuse to go buy them.

Every Amiga user should own a spreadsheet, and MaxiPlan rates as a "best buy". The program has "Amiga" stamped all over it. The new revision takes care of most of the problems, and the enhancement due around year-end should make this one of the best programs available for any computer. But don't wait for the enhancement - go buy MaxiPlan right now and take advantage of the power of this program, and your Amiga, today.

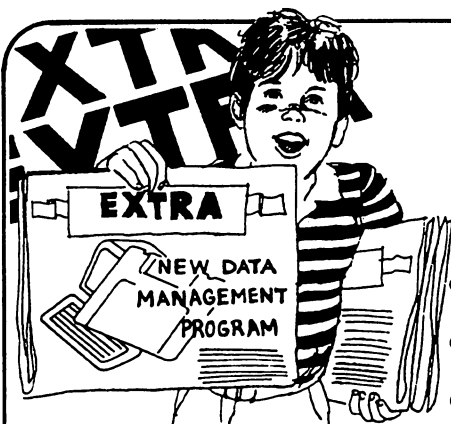
MaxiPlan

\$150.00 + \$25.00 for MaxiPlan Macros
((\$50.00 after introductory period))

Maxisoft

2817 Sloat Road
Pebble Beach, CA 93953
408-373-4018
800-942-6294

•AC•



INFO+

The Personal Database Management System

- EXTRA _ Easy to use. No new language to learn.
- EXTRA _ On line help. Just press for command information.
- EXTRA _ Flexibility, with pull down menus, automatic filing and field sizing.

EXTRA _ LOW price...

JUST \$49.99 !!!



Order NOW! (305) 657-4355

**Eastern Telecom Inc.
9514 Brimton Drive
Orlando, FL 32817**

**FREE Shipping in Continental U.S.
Dealer Inquiries**

EXTRA APPLICATIONS

Electronic Files
Student Grades
Monthly Bills
Inventory
Birthdays
Configurations
Rental Property
Shopping Lists
Book Titles
Phone Lists
Video Tape

EXTRA FEATURES

Fast Execution
Multi-Tasking
Dynamic Positioning
Sorts Any Field
All Screen Colors
Mouse-Key Entry
Numeric Pad
Formatted Printing
Variable Fields
32765 Records
Not Copy Protected

EXTRA USEFUL

INFO+ helps you create, update, and retain computerized files of all your valuable information. You can sort the data by date, size, name, cost, etc. INFO+ is an automatic secretary and file cabinet all in one. Taking a vacation or buying a car? Let INFO+ do the analysis for you. Never forget a recipe or birthday again. INFO+ is the ideal reminder system, and it has a large memory.

Gizmoz

"Digital Creation's collection of eighteen clever little programs which handle a myriad of different tasks."

Reviewed by
Peter Wayner

Gizmoz is Digital Creation's collection of eighteen clever little programs that fill up the memory while handling a myriad of different tasks from the menial projects of address and phone number filing to the cloak and dagger luxuries of encryption as well as plenty of other useful things. These programs are meant to imitate and go beyond the IBM-PC industry standard, Sidekick the memory resident program that brought the illusion of multi-tasking to the IBM. In the years since its introduction in 1984, Sidekick and its imitations have filled the business world's computers with programs that wait in high memory for specific keystrokes to call forth appointment calendars, calculators, notepads and telephone directories. It is not surprising, then, that programs that do these very same things are appearing for the Amiga, a computer with real multi-tasking capability. Gizmoz will do much more than Sidekick and its companion product SuperKey combined and at a fraction of the price. It is also not copy-protected.

In a nutshell, Gizmoz will keep appointments, phone numbers and memos straight, as well as print them out for storage in the black binder that comes with the package. It will also serve as a calculator, encrypt files and set up keyboard macros. To do just this much, Borland International, the previous king of good, inexpensive products charges \$175 for Sidekick and Superkey.

Not being one to stop a good thing, Digital Creations also included a terminal program, an animated cuckoo clock, an announcer, a graphing program and a Life game. And for the programmer, there are pop-up reference cards for AmigaDOS and ABasic, a file compression program, a graphic memory display and a tool for setting the priority of the various jobs.

These tools, productivity assistants and toys take up so much room on the disk there is no room for any of the workbench files. Since I only have one drive, I copied several programs at a time onto a workbench disk to test them. This saves disk swapping whenever the programs want to access the device routines. Ideally, I would want everything on a harddisk, but until one appears next to my Amiga, I put some programs on my writing disk, some on my programming disks and the rest on a games disk.

For writing and general notekeeping, I placed the calendar, rolldex and memopad on one disk. They are probably the most productive of the programs in the collection if you use your Amiga for office work.

The Memopad is an excellent, little editor that uses the mouse interface to handle any text file. Several of these Memopad

windows can be opened at once, and anything cut or copied from one file can be pasted into another. It makes such a good, general-purpose editor that a friend of mine has renamed his ED and placed it in the C directory of all his Workbench disks. Its only problem is that it will scroll to handle lines up to 256 characters long. I know some would call this a feature, but the editor is so convenient, and small I would like to use it to write. (It scrolls smoothly and quickly, unlike Textcraft.)

The Rolldex program is a digital imitation of a cardfile that will dial the phone numbers if a Hayes compatible modem is attached. There are no specific name or address fields because each card is like a small Memopad with fixed boundaries. The editing commands are the same, but there is no scrolling in any direction. This means that the cursor must be placed over the phone number before the dial routine is called, a pain that can be avoided by putting the phone number in the upper-left-hand corner of each card where the cursor appears initially. I would rather there was one specific field for phone numbers. It would make things a bit easier.

The Calendar is the last of the three main office assistants, and it is just as good. Every day from January 1, 1900 to December 31, 2099 has its own 50 line, fixed boundary memopad. In addition, if a line describing an appointment begins with an asterix, the program will issue a beep or a flash at the appropriate time if the Calendar is open on the Workbench. Several different calendars can be stored separately on the disk if there is any reason to keep things separate. If you work enough on the Amiga, it would be worth the trouble of synchronizing the clock everyday to get these reminders.

All three of these office assistants are designed to produce files that can be read by the program named Blackbook. This program will format the files and print them out on 8 1/2 by 11 paper. These pieces must be folded in half and punched with holes before they are put in the black binder. This makes the binder smaller than a regular one, but I would rather use a normal binder and save the trouble of folding.

The package also comes with different calculators for scientists, programmers and financial analysts. I split them up onto different disks, but I will discuss them together since there are so many similarities. Each of them will handle the rudimentary four functions as well as display a tape history of the calculations along side the calculator face. This tape can also be saved to a file or printed out on the printer. This is a very handy feature. Either the mouse or the keypad can be used to operate the various keys.

DIGITIZED INSTRUMENTS FOR INSTANT MUSIC OWNERS



Here is what you have been waiting for! This disk is full of digital samples of **real** instruments that can be played by your instant music® program at **HI FI** quality.

- Write & play songs that sound real
- Use in existing songs or use to write your own
- I.F.F. compatible
- Over 30 instruments
- Classical, Contemporary & Rock instruments
- Can be used by 1 drive systems
- Instruments can be copied to other song disks
- Includes guitars (4 types), drums, organs, trumpets, flute, violin, etc.

\$20 POSTPAID

(Connecticut Residents Add \$1.50 Sales Tax)



Actionware—David Crane

1039 Farmington Ave.

West Hartford, CT 06107

(203) 233-0151

Instant Music is a trademark of Electronic Arts.



The scientific version will calculate the standard trigonometric functions, logarithms, exponentials in any number of decimal places. The calculator handles exponents up to 99 and carries 15 digits of accuracy through its internal calculations. It is equivalent to a Texas Instrument's TI-30.

I put the programmer's calculator on my programming disk, even though I don't know when I'll begin needing to compute arithmetic shift, logical shift, rotate, OR, AND, XOR and NOT in either binary, octal, decimal or hexadecimal. Each number entered into the machine can be considered to be 1, 2, 4, 8, 16 or 32 bits long. This makes a difference since negative numbers are presented as two's complements. Since 32 bits of a binary number don't fit on the calculator window, there is a provision to switch between blocks of digits. This is cumbersome, though, and it would have been better if the calculator window expanded.

The financial calculator has an equally large array of buttons that make any computations involving money, time and the compounding of interest very easy. It will do everything Electronic Art's Financial Cookbook will do, but with absolutely no handholding for the uninitiated. Unfortunately, the errors I found in the handbook were in this section. Two of the examples for testing the calculator were wrong. They were typographical errors that made me wonder if my calculator was screwed up. A phone call to Digital Creations assured me that since the calculator worked correctly, the manual must be wrong. There is also one

undocumented feature of the calculators that allows you to change the constant. I had trouble getting this to work correctly, but I think it is because I use HP calculators. These are programmed to work like Texas Instrument's straight-forward system.

One of drawers is entitled Audio-Visual and I moved all of it's programs to the toy disks. One of these toys is an announcing program that will translate any English text into phonemes and speak them over the sound channels. Just as in BASIC, the cadence, pitch, inflection, sampling rate and sex of the voice can be adjusted. There is also a neat animated face that changes the size of it's rectangle mouth in time with the words. This little toy can be put to some practical use by programming batch files to use it to say, "The compiler is done."

The other pure toy in the drawer is a cuckoo clock that has a little animated bird call out the hour. It is a nice job of animation, but little more than a novelty. The ticking becomes so annoying I usually shut off the sound.

I suppose Life is considered a toy by most people, but Computer Scientists would think otherwise. A system of little cells can be built complex enough to do simple calculations. This implementation helps study the theoretical application by providing a library of the most important Life forms like the glider gun. This crazy little pattern will regularly spit out another pattern called the glider that scoots across the screen. It is used theoretically to simulate a computer's clock. There are many other interesting forms, and it makes this particular implementation much better than the usual version.

The last program I moved over is the graphing program. It will take up to twelve numbers and produce a pie or a bar graph in four different colors with three different shading options. Unfortunately, it leaves the table of data on the side of the screen.

While this makes it easy to change things, it is impossible to take a photograph directly from the screen. Since I don't have a color printer, I think this ability is a necessity.

Everything so far, would have been more than enough for a regular software package, but there is still another drawer included entitled Accessories. In it, the programmers at Digital Creations placed all of the extra goodies they couldn't fit elsewhere. I'm still not sure where they will end up on my disks.

The first interesting little program is called POPUP. It creates a window that will slide up and down on top of the Workbench windows. Gizmoz contains two different files filled with reference data on AmigaDOS and AmigaBASIC to be loaded by POPUP. The information provided is somewhat cursory, but this cuts down on the memory that is gobbled up. It is, though, a great deal more documentation on the DOS than Commodore provided. If anyone isn't satisfied, they can use POPUP with any text file and edit their own references.

Another interesting gadget, albeit without any obvious applications, is the graphic memory display program. This is another POPUP-like window that contains a box filled with patches of different colors. One color is allocated memory, the other is free memory. Each dot on the screen represents an eight byte block. It is really quite interesting to see how the Amiga will

tend to leave tiny chunks free all over the place. I guess multi-tasking memory management is never perfect. The program will store a "snapshot" of the memory allocation at a particular time and then XOR it with the current picture on command. This makes it quite easy to see how a specific program fills up the memory.

Another of these interesting tools without an obvious application is the SetPriority program. This is the perfect gift for the child that can't keep his hands off things he is not supposed to touch. I have crashed the machine almost every time I tried resetting the priorities of the various tasks in the queue. It is quite easy to set a computation intensive program higher than the input console and prevent the computer from looking for mouse movement or break commands. The most success I had with it was setting the priority of the memory display program to the maximum to watch a very dynamic view of the machine's memory allocation. Usually the graph of the memory is only updated when the machine gets around to it.

If anyone uses a modem, they will be happy to note that a simple terminal program is included. It will emulate the standard terminal protocols like the ADM-3 and the VT100 while providing the options of stripping line-feeds and saving the text to disk. Unfortunately, there is no facility for uploading a file from the Amiga. This is probably because Digital Creations offers a separate communications package.

In this drawer, there are three handy programs for text processing. One is called HotKey and it reprograms the keyboard to replace individual keys with long strings. With it in place, everytime control, alt and the key are pressed, HotKey will intervene and the

computer will receive the long, pre-programmed string. It is quite useful whenever long names are used repeatedly. Dostoyevsky would have loved it.

Rounding out the collection are two programs that help with text files that lie around on disks. One compresses files using the standard Huffman coding approach and the other encrypts them with Digital Creation's own algorithm. People knowledgeable about cryptography would want to know that it is a version of a Vernam cipher that uses the key as a seed for a pseudo-random sequence. This means it is moderately secure for most applications. Anyone using the encryption algorithm would be well advised to use the file compression program before encrypting. This makes it much more difficult to crack the cipher.

As a final note, the buyer should be warned that some of the programs can be garnered from the public domain. For instance, the first AMICUS disk contains a close copy of the announcing program complete with animated face. There is also a file compression program that yields almost exactly the same percentage of compression on the same disk. Simple terminal packages, Life programs, encryption schemes and graphic free memory displays are also available from AMICUS. This does not mean the collection is not a good value. Every one of the programs works and is well documented. Each program can be accessed from both the CLI and the Workbench and the editing commands are the standard for every program that handles text. These may be luxuries to some people who love the public domain, but I think Gizmoz is a good bargain for people who like to have a software with the finishing touches.

•AC•

**MEGAPORT
COMPUTER CENTER**

Announces New Business Software for the Amiga

Amiga Cash Register \$99.95

- *Point and Click operation
- *Uses Amiga Intuition
- *Up to 30 Salespeople 1,000 items
- *Prints
 - Invoice
 - Cost of Sales
 - Sales Tax Report
 - Sales Analysis

Houston Inst. 695 Plotter Driver

for Aegis Draw™

only \$29.95

CALL FOR DEALERS AMIVAR

We have turn key
proven packages
for:

- Mortgage Processing
- Realestate Rental Management
- Pool Estimating
- Carpet Store Point of Sale
and Inventory Management
- Real Estate Appraisal

THE Amiga Dealer in Florida

MEGAPORT COMPUTER CENTER

The Friendly Computer Store With The Friendly Computer

1209 U.S. 41 By-Pass So.

Venice, FL 33595

(813) 484-4787

Amiga is a trademark of Commodore-Amiga, Inc. Aegis Draw is a trademark of Aegis Development

(Dealer Inquiries Welcome)

Attention Amiga Owners

You should know the *Side-Effects* of owning your Amiga!!!

Side-ARM (Amiga Resource Module):

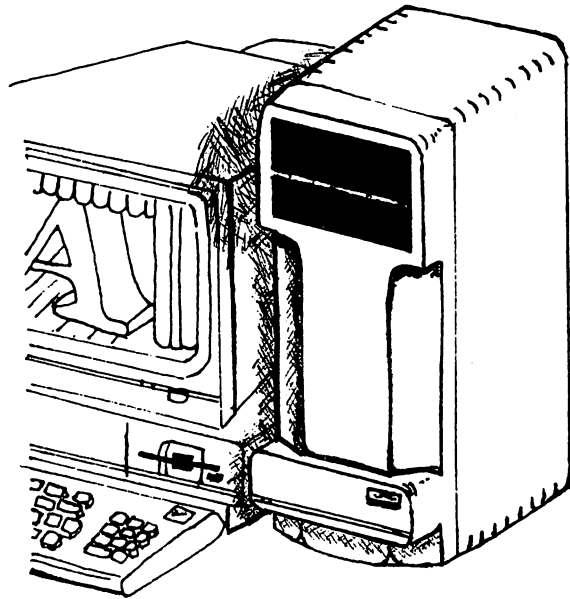
- 6/12 slot back plane
- optional 86 pin bus pass-through
- space for 2 half-height drives
- complete with standard power connectors

Side-Store (memory card):

- 2 megabytes per card
- no wait states
- RAM-disk that survives resets/reboots

Side-Track (Disk & Clock):

- 20, 30, and up to 150 megabytes
- Reed-Soloman error correction code
- ST-506 compatible
- Battery backed real-time clock



Side-ARM

Side-Band (Midi Interface, music synthesiser)

Side-Port (serial, parallel, SCSI)

Side-More.....

The following apply to all items:

- fully Zorro bus compatible
- auto-config standard
- matching plastic cover
- burned-in before shipping
- 6-month warranty

S

I

D

EFFECTS INC

6513 Johnsdale Rd.
Raleigh, NC 27615
Voice: (919) 876-1434
BBS: (919) 471-6436

The Loan Information Program

"...all the necessary information about that loan before you get serious with the sales person."

by Bryan Catley

I don't know about you, but when I'm in a situation that could very well end with the need for a loan, (buying a car, a house, or refinancing a house, etc.), I want all the necessary information about that loan before I get serious with the sales person. The monthly payments are of vital interest, especially those surrounding the "most likely" rate because rates do tend to fluctuate. Also of importance is knowing how much of each payment will go towards paying off the interest, and how much will go towards the principal. "The Loan Information Program" will provide you with this information, and show-off some of the Amiga's special features at the same time.

Carefully type in the accompanying program and remember to save a copy on disk before trying to execute it. In fact, it would be wise to save the program several times as you enter it.

Using the Program

When you run "The Loan Information Program", you will be greeted orally, and presented with the Title Screen. There will be four boxes on the screen, labelled from left to right "Help", "Payments", "Interest", and "Quit". Select the desired function by "clicking" in the appropriate box.

Help

The help option provides three successive windows of help information. The first describes the "Payments" option, the second the "Interest" option, while the third offers a few words about the printer routines. You have the option of terminating "Help" with each window.

Payments

The payment option will allow you to determine the monthly payment for a specified loan. When selected, you will be presented with a "Requester" which will prompt you in sequence, for the amount of the loan, the interest rate, and the length of the loan in years. You must press RETURN after entering each item.

When they have all been entered, "OK" and "Cancel" boxes will appear in the Requester. If you select "Cancel", you will be returned to the Title Screen. If "OK" is selected, you will be presented with a table detailing the monthly payments for loans of amounts surrounding the specified amount, at various rates of interest surrounding the specified rate. The column and row representing the specified values will be highlighted. This table makes the comparison of various loan amounts, at varying rates of interest, very easy.

When ready, click the mouse in the desired box; "Interest" will take you directly to the Interest screen, "Print" will produce a hard copy version of the information on the screen, on a printer attached to the parallel port.

Interest

The interest option will show you the amounts of interest and principal paid by month, (with totals) for any desired 12 month period during the life of the loan. When selected from the Title Screen, one of two things will happen. If you have previously used the "Payments" or "Interest" options, you will be asked if the same loan information is to be used. If you then select "No", or if you did not use these options previously, you will be requested to enter the same information as is required for the "Payments" option.

When the correct loan information has been determined, the program takes a couple of seconds and constructs an array of information regarding every payment for the life of the loan. It then presents you with a screen containing interest paid, principal paid, balance, and totals of interest and principal paid, for months one to 12 of the loan.

Across the top of the screen will be seven boxes (or gadgets) which you may select as desired, and as appropriate:

First Year

Provides information for the first year of the loan. The number of payments shown defaults to 12 when "Interest" is first selected, but you may change the number at will via the "Change" gadget. This option is automatically selected when "Change" has been used.

Last Year

Provides information for the payments remaining in the last year of the loan. The number of payments shown will be 12, or 12 less the number of payments in the first year of the loan.

Next 12

This will cause information for the next 12 months to be listed. If the highest current month plus 12 is beyond the life of the loan, "Last Year" will automatically be selected.

Prev 12

This will cause information for the previous 12 months to be displayed. If the lowest current month less 12, is less than one, "First Year" will be automatically selected.

CARDINAL ANNOUNCES ITS

2 x 2

EXPANSION DISK DRIVE

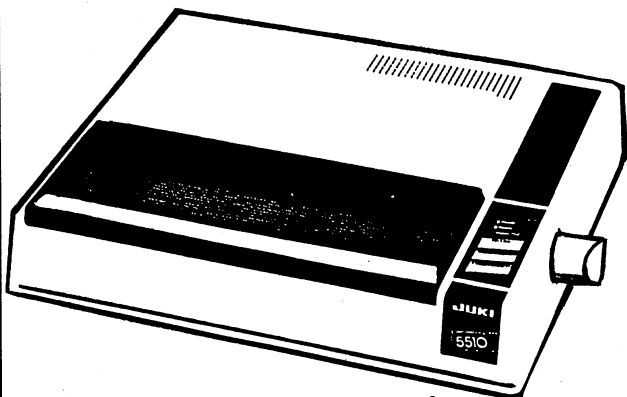
GIVE YOUR AMIGA
2.6 MEGABYTES

2 x 2 CONSISTS OF:
2 5 1/4" 80 track drives,
electronics and software.

- AMIGA DOS MODE - Emulates 3.5" drives with 880K each.
- PC DOS MODE - Provides dual 40 track, 360K drives.

\$595.00

JUKI 5510 C
Color Printer
(Uses Epson Codes)
\$469.00



Shipping extra!

Cardinal Software 14840 Build America Drive
Woodbridge, VA 22191 Info: (703) 491-6494
Order now! (10-4 Mon-Sat)



800 762-5645



Change

When the interest option is initially invoked, the program assumes the first year of the loan will contain 12 payments. However, this is usually not the case. By selecting this gadget you may set the number of payments for the first year to whatever is appropriate.

Print

Produces a hard copy version of the information on the screen, on a printer attached to the parallel port.

OK

Selecting this gadget will terminate the "Interest" option and return you to the Title Screen.

Quit

This option will terminate the program and return you to the Amiga Basic Output Window.

Programming Notes

It is hoped that the following notes will help you in obtaining a better understanding of "The Loan Information Program" and how it does what it does.

Information Producing Formulas

The formulas used in this program should produce accurate results. None-the-less, it would probably be wise to double check results with other sources before basing any major financial decisions on them!

Of Gadgets and Mice

This program contains 21 different gadgets for the user to "click" in, with up to seven gadgets being used on one screen at the same time! To include the code to draw each individual gadget as required, and to include the mouse checking code for each individual gadget would really have lengthened the program and made it unwieldy. The simplification process: decide on a common format for the gadgets, and place all information about them in two arrays. These arrays will, in turn, be used by one gadget drawing routine, and one mouse checking routine. The program thus becomes smaller, and changes or additions to the gadgets become much easier.

The common format decided on for this program was one line of text in a box surrounded by a second box which is two pixels larger in each direction. When the gadget is selected, the background color in the inner box, and the text color are reversed for as long as the user holds the left mouse button down. This provides a "flashing" effect. Additionally, each box has a "shadow" on the right and lower sides. The first of the two arrays contains, for each gadget:

- x coordinate of upper left hand corner
- y coordinate of upper left hand corner
- length in pixels
- height in pixels
- background palette color
- foreground palette color
- shadow palette color

The second contains the text to be placed in each gadget.

Gadgets are drawn via their relative locations within the array. For example, to draw the fifth and sixth gadgets in the Help window, the commands used are:

```
boxx=4:boxy=5:GOSUB DrawGadgets
```

(remember, the first entry in an array is entry zero).

The "GetGadget" subroutine is entered with the variables "boxx" and "boxy" set to the lower and upper box numbers to check. Upon exit, the variable "type" contains a value from one up to the number of boxes being checked. This allows it to become the object of an IF or ON statement to direct subsequent program flow. For example, after drawing boxes five and six as described above, the "GetGadget" subroutine might be used as follows:

```
boxx=4:boxy=5:GOSUB GetGadget
ONH Htype GOTO ....
```

When first entered, "GetGadget" waits for the left mouse button to be pressed. When it is, the "x" and "y" coordinates are extracted and compared against the specified gadgets. If a match is found, foreground and background colors are reversed and the inner box of the gadget is re-drawn (causing it to "flash"), and the variable "type" is set as appropriate. The subroutine then waits for the user to release the button. It then either repeats the process (if no match was found) or returns to the user after re-drawing the gadget in its normal colors.

Speech

It must be admitted, in all honesty, that speech as used in this program, is mostly gimmick! But it is available, so why not use it? Besides, it does provide oral instructions and comments as the program decides it appropriate. The one important thing to note is that if you are going to use speech in your programs, include a SAY statement at the very beginning of the program, even if nothing is said: **SAY TRANSLATE\$("")**. This is important because Basic does not keep the speech routines in memory, it loads them from the Workbench disk when they are first needed. Now, if the Workbench disk is not in the internal drive, the system will display a Requester for it on the system screen, which means that if you've already created your own screens and/or windows, it may be hidden from view, and the user may very well end up thinking the system has crashed. Including a SAY statement at the very beginning of your program avoids this problem.

Retrieving User Input

When you run the program, you will undoubtedly notice that when entering information about the loan, the "OK/Cancel" gadgets do not appear until all the required information has been entered. This is because "LINE INPUT" is used to retrieve the information, and once it receives control, it will not relinquish it until the user presses the RETURN key. This makes it impossible to check the condition of the mouse during this time; hence, the gadgets do not appear until all information has been entered.

Printer Routines

To provide hard copy of the selected screen, the printer subroutine makes use of horizontal spacing, vertical spacing, form feeding, double width printing, and emphasized printing; all in

Conversation With A Computer

(It's a lot of fun, a brain teaser and a programming guide too!)

"Very highly recommended by me is Conversation With A Computer, from Jenday Software, a set of games and conversation written in Amiga Basic, and shipped with the source code provided. It is entertaining, amusing, thought provoking, and just plain fun. If you have any interest in programming in BASIC on the Amiga, this is a must have for the examples."

—MATTHEW LEEDS, Commodore Microcomputers

This program really shows off Amiga's talents: lots of color graphics, mouse routines, voice synthesis, sound and animation. The 2,000 lines of Amiga Basic can be listed to screen or printer. The documentation describes in detail, module by module, how it all works. There is a coded example of virtually every one of Amiga Basic's powerful features.

You'll be challenged to three mind games. Memory Test will drive you to drink. Battle of Numbers and Pegboard are two of the most elegant logic games of all time. It's your brain against Amiga's silicon!

The program is professionally packaged with comprehensive typeset documentation. It requires 512K. It is not copy protected. Now includes an introduction to the C language.

JENDAY
SOFTWARE

P.O. Box 4313
Garden Grove, CA 92642

All orders are shipped by first class
mail within 24 hours of receiving
your personal check or money
order.

DEALER INQUIRIES INVITED

(714) 636-3378

ORDER FORM

Please rush me Conversation With A Computer with source code.
Enclosed please find a check for \$29.50 plus \$2.50 postage and
handling. (Californians add \$1.77 sales tax.)

Name _____
Address _____
City _____
State _____
Zip _____

addition to "normal" printing! To do this, extensive use is made of printer control codes and, unfortunately, these vary from printer to printer. As written, the program will print correctly on Epson, and Epson compatible dot-matrix printers. If you have any other kind of printer, you may receive some strange results! Should this occur, you will need to change the control codes that have been used to ones which are acceptable to your printer. To do this, find the necessary codes in your printer instruction manual and substitute as necessary. Each printer function is only invoked from one place within the program, so it will not be necessary to search high and low!

Well, I'm sure this program will prove useful to you. Both from the point of view of being a useful program to have in your home program library, and also as an example of how to use some of the features available in Amiga Basic. Enjoy it!

```
' The Loan Information Program
' Version 2.0
' June 1986
```

```
' Bryan D. Catley
' 1239 Portner Road
' Alexandria VA, 22314
```

```
' No warranty, explicit or implicit, is given regarding the
' accuracy of the information produced by this program.
```

```
CLEAR:DEFDBL p:numbx=21
amtinc=2000:itrinc=.25:maxyrs=40:sw=0
n=0:m=0:amt=0:itr=0:rate=0:rateinc=0:ratefact=0
yrs=0:mths=0:pay=0:req=0:mthlst=0:type=0
mths$="" :yrs$="" :itr$="" :mthlst$=""
Pnum=1:PCount=0:Fval=0:boxx=0:boxy=0
DIM bx (numbx-1,6),bxtxt$(numbx-1),voice$(8)
DIM mthly(14,6),tabx((maxyrs*12)+1,3)
FOR n=0 TO 8:READ voice$(n):NEXT
DATA 110,0,150,0,22200,64,10,1,0
SAY TRANSLATE$("Hi, Welcome to the Loan Information
Program."),voice$
GOSUB BldGadgets
' Main Title
DATA 64,60,80,16,0,2,7," Help"
DATA 208,60,80,16,0,1,7,"Payments"
DATA 344,60,80,16,0,1,7,"Interest"
DATA 488,60,80,16,0,4,7," Quit"
' Help Window
DATA 16,116,64,16,7,2,0," More"
DATA 148,116,72,16,7,4,0,"End Help"
DATA 16,116,64,16,7,2,0," OK"
' Payments
DATA 204, 12,86,16,7,1,0," Interest"
DATA 292, 12,78,16,7,1,0," Print"
DATA 372, 12,78,16,7,2,0," OK"
' Requester
DATA 16,100,64,16,7,2,0," OK"
DATA 160,100,64,16,7,4,0,"Cancel"
' Same Loan?
DATA 24, 44,40,16,7,2,0,"Yes"
DATA 104, 44,40,16,7,4,0," No"
' Interest
DATA 100, 12,86,16,7,1,0,"First Year"
DATA 188, 12,78,16,7,1,0,"Last Year"
DATA 268, 12,62,16,7,1,0,"Next 12"
DATA 332, 12,62,16,7,1,0,"Prev 12"
DATA 396, 12,54,16,7,1,0,"Change"
DATA 452, 12,46,16,7,1,0,"Print"
DATA 500, 12,38,16,7,2,0," OK"
MTa=0:MTb=3:Ha=4:Hb=5:Hc=6:Ya=7:Yb=9
Ra=10:Rb=11:SLa=12:SLb=13:Ia=14:Ib=20
SAY TRANSLATE$("")
```

```
Main:
Logo80 3
boxx=MTa:boxy=MTb:GOSUB DrawGadgets
```

```
Main1:
COLOR Mag,Blk:LOCATE 13,39:PRINT"THE"
LOCATE 15,37:PRINT"L O A N"
LOCATE 17,30:PRINT"I N F O R M A T I O N"
LOCATE 19,37:PRINT"PROGRAM"
IF First=0 THEN
  First=1
  SAY TRANSLATE$("Please click in the desired box to
continue."),voice$
END IF
Main2:
boxx=MTa:boxy=MTb:GOSUB GetGadget
ON type GOTO HelpRoutine,PayRoutine,IntRoutine,QuitRoutine
```

```
HelpRoutine:
WINDOW 3,,(40,16)-(280,152),0,1
COLOR Blk,Yel:CLS:LOCATE 2,1
PRINT"Selecting 'Payments' will"
PRINT"allow you to examine a range"
PRINT"of loan amounts, at various"
PRINT"rates of interest, around the"
PRINT"values you supply."
PRINT" "
PRINT"Information specific to your"
PRINT"request is high-lighted."
PRINT" "
PRINT"Do not include commas when"
PRINT"entering dollar amounts."
boxx=Ha:boxy=Hb:GOSUB DrawGadgets
boxx=Ha:boxy=Hb:GOSUB GetGadget
IF type=2 THEN EndHelp
COLOR Blk,Yel:CLS:LOCATE 2,1
PRINT"Selecting 'Interest' will"
PRINT"provide you with the amounts"
PRINT"of interest and principal"
PRINT"paid, month by month, for any"
PRINT"12 month period during the"
PRINT"life of the specified loan."
PRINT" "
PRINT"Short first and last years"
PRINT"are handled, and you may page"
PRINT"through the 12 month periods."
PRINT" "
PRINT"Do not include commas when"
PRINT"entering dollar amounts."
boxx=Ha:boxy=Hb:GOSUB DrawGadgets
boxx=Ha:boxy=Hb:GOSUB GetGadget
IF type=2 THEN EndHelp
COLOR Blk,Yel:CLS:LOCATE 2,1
PRINT"The printer routine is set up
PRINT"for dot-matrix EPSON or EPSON"
PRINT"compatible printers. All"
PRINT"printing is done from one area"
PRINT"of the program, allowing easy"
PRINT"modifications if appropriate."
boxx=Hc:boxy=Hc:GOSUB DrawGadgets
boxx=Hc:boxy=Hc:GOSUB GetGadget
EndHelp:
COLOR Blk,Cyn:WINDOW CLOSE 3:GOTO Main1
```

```
PayRoutine:
req=1:GOSUB Requestor1:IF type=2 THEN Main2
amtinc=2000:IF amt<amtinc*4 THEN amtinc=amt/4
amt=amt-(amtinc*3)
itr=itr-(7*itrinc):rate=itr/1200
rateinc=(itrinc/100)/12
COLOR ,Blk:CLS
LINE(40,8)-(600,184),Cyn,bf
col=0
FOR x=4 TO 188 STEP 8
  col=col+1:IF col>7 THEN col=1
  LINE(0,x)-(27,x),col
  LINE(608,x)-(631,x),col
NEXT
LOCATE 1,24:COLOR Mag:PRINT"LOAN REPAYMENT COMPARISONS, BY
MONTH"
boxx=Ya:boxy=Yb:GOSUB DrawGadgets
COLOR Blu,Cyn:LOCATE 5,15
PRINT"<===== ";yrs$;
PRINT" Year Loan Amounts =====>"
LOCATE 7,7:PRINT"Rates":COLOR Wht
AREA (48,112):AREA (320,112):AREA (320,40):AREA (376,40)
AREA (376,112):AREA (592,112):AREA (592,119):AREA (376,119)
AREA (376,176):AREA (320,176):AREA (320,119):AREA (48,119)
AREAFILL:COLOR Blk
```

```

FOR n=0 TO 6
  COLOR ,Cyn
  IF n=3 THEN COLOR ,Wht
  LOCATE 6,15+n*9:PRINT USING "#####";amt+(n*amtinc)
NEXT
FOR n=0 TO 14
  COLOR ,Cyn
  IF n=7 THEN COLOR ,Wht
  LOCATE 8+n,7
  PRINT USING "###.###";itr+(n*itrinc)
  ratefact=rate+(n*rateinc)
  FOR m=0 TO 6
    COLOR ,Cyn
    IF n=7 OR m=3 THEN COLOR ,Wht
    pay=(ratefact/(1-(1/(1+ratefact)^mths)))*(amt+(m*amtinc))
    pay=INT(pay*100+.5)/100
    IF pay<10000! THEN
      LOCATE 8+n,14+m*9:PRINT USING "###.###";pay
      mthly(n,m)=pay
    ELSE
      SAY TRANSLATE$("That's a big loan!"),voice$
      COLOR Red:LOCATE 3,23
      PRINT"Payments are too large to display"
      COLOR Blk:m=6:n=14
    END IF
  NEXT m
NEXT n
PayRtnWhat:
boxx=Ya:boxy=Yb:GOSUB GetGadget
ON type GOTO DoInt0,PayPrint,Main

PayPrint:
GOSUB POpen:GOSUB NewPage
Pval=21:GOSUB PHorSp:GOSUB PBlidOn
PRINT #Pnum,"LOAN REPAYMENT COMPARISONS, BY MONTH":GOSUB
PBlidOff
PRINT #Pnum," "
Pval=13:GOSUB PHorSp
PRINT #Pnum,"<----->";yrs$;
PRINT #Pnum," Year Loan Amounts <----->"
Pval=10:GOSUB PHorSp
FOR n=0 TO 6
  IF n=3 THEN GOSUB PBlidOn
  IF n=4 THEN GOSUB PBlidOff
  PRINT #Pnum,USING" #####";amt+(n*amtinc);
NEXT:PRINT #Pnum," "
Pval=5:GOSUB PHorSp:PRINT #Pnum,"Rates"
FOR n=0 TO 14
  IF n=7 THEN GOSUB PBlidOn
  IF n=8 THEN GOSUB PBlidOff
  Pval=5:GOSUB PHorSp:PRINT
  #Pnum,USING"###.###";itr+(n*itrinc);
  FOR m=0 TO 6
    IF m=3 AND n<>7 THEN GOSUB PBlidOn
    IF m=4 AND n<>7 THEN GOSUB PBlidOff
    PRINT #Pnum,USING" #####";mthly(n,m);
  NEXT:PRINT #Pnum," "
NEXT
Pval=4:GOSUB PVrtSp:GOSUB PClose
GOTO PayRtnWhat

IntRoutine:
IF mths=0 THEN DoInt1
WINDOW 5,,(72,40)-(240,120),0,1
COLOR Blu,Yel:CLS:LOCATE 2,2
PRINT"For the same loan?"
boxx=SLa:boxy=SLb:GOSUB DrawGadgets
boxx=SLa:boxy=SLb:GOSUB GetGadget
WINDOW CLOSE 5:COLOR Blk,Cyn
ON type GOTO DoInt0,DoInt1
DoInt0:
IF req=1 THEN amt=amt+(amtinc*3):itr=itr+(7*itrinc)
GOTO DoInt2
DoInt1:
req=2:GOSUB Requestor1:IF type=2 THEN Main2
DoInt2:
sw=0:ratefact=itr/1200
COLOR ,Blk:CLS
LINE(80,8)-(560,184),Cyn,bf
col=0
FOR x=4 TO 188 STEP 8
  col=col+1:IF col>7 THEN col=1
  LINE(0,x)-(72,x),col
  LINE(568,x)-(631,x),col
NEXT
COLOR Mag,Blk:LOCATE 1,22

```

"C" Programmers and Developers "IntuiSeeds" Intuition Application Library

Features:

- Easy Building of Screens, Windows, Requesters, Menus and a full complement of Gadgets and Images.
- Window Management - includes the option to monitor all windows through a single IDCM Port.
- Management of pointers for Screen, Windows and Requesters.
- Automatic linking and updating of Gadgets to Windows and Requesters.
- Memory Management through the use of a single routine.
- Full documentation for all routines.

To order, send \$69.95 (US) check or money order (plus \$2.50 S/H) to:

GreenThumb Software
23 Dennison Ave.
P.O. Box 2949
Binghamton, NY 13902

Intuition is a trademark of Commodore-Amiga
IntuiSeeds is a trademark of GreenThumb Software

GreenThumb Software
"Seeds for the Creative"

```

PRINT"INTEREST & PRINCIPAL PAYMENTS BY MONTH"
LOCATE 3,33:COLOR Red,Cyn:PRINT"Please Standby"
pay=(ratefact/(1-(1/(1+ratefact)^mths)))*amt
pay=INT(pay*100+.5)/100
tabx(0,0)=amt:tabx(0,1)=0:tabx(0,2)=0
FOR n=1 TO mths
  tabx(n,1)=INT((tabx(n-1,0)*ratefact)*100+.5)/100
  tabx(n,0)=tabx(n-1,0)-pay+tabx(n,1)
  tabx(n,2)=pay-tabx(n,1)
NEXT
LOCATE 3,33:PRINT SPACE$(14)
boxx=Ia:boxy=Ib:GOSUB DrawGadgets
LOCATE 5,21:COLOR Blu,Cyn
PRINT"Month Interest Principal Balance"
LOCATE 20,29:PRINT"Loan Amount:"
LOCATE 21,27:PRINT"Interest Rate:"
LOCATE 22,25:PRINT"Number of Years:"
LOCATE 23,25:PRINT"Monthly payment:";
COLOR Blk
LOCATE 20,42:PRINT USING "#####";amt
LOCATE 21,42:PRINT USING "###.###";itr:PRINT"&"
LOCATE 22,42:PRINT USING "###";yrs;
PRINT" (:;PRINT USING "#####";mths:PRINT" months)"
LOCATE 23,42:PRINT USING "###.###";pay;
mthlst=12:type=1
DoRequest:
ON type GOTO
Year1,Lastyr,Plus12,Minus12,DoChange,IntPrint,IntEnd
Year1:
mthl=1:mthend=mthlst:GOTO DoList
Lastyr:
IF mthlst=12 THEN
  mthl=mths-11
ELSE
  mthl=mths-(11-mthlst)
END IF
mthend=mths:GOTO DoList
Plus12:
IF mthend+12>mths THEN Lastyr
mthl=mthend+1:mthend=mthl+11:GOTO DoList

```

Use Your Own Photos

...in programs such as Deluxe Paint or Images. Your pictures from flat art 2" x 3" to 8½" x 11" or color slides 35mm to 4" x 5" will be digitized by the Digi-View system to 32 color, 320 x 200 resolution pictures compatible with any IFF paint program. Minimum order is 8 images for \$24, disk included (California residents add state sales tax) plus \$2.50 shipping. Additional images \$2.00 each. Pictures may be cropped to fill the screen. For no cropping specify *full frame*.

Photographic Clip Art!

Sample disk includes landscapes, clouds, trees, buildings, celestial object, etc. Use in your own IFF paint programs. Customize to suit your needs by flipping, stretching, stamping, changing colors. More realistic than drawings! Order Clip Art Sampler #1. Catalog of other Clip Art disks will be included. \$20 (California residents add state sales tax) plus \$2.00 shipping.

DIGI-PIX

800 Heinz Street □ Berkeley, California 94710 □ (415) 644-0614

Deluxe Paint is a registered trademark of Electronic Arts. Images is a registered trademark of Aegis Software. Digi-View is a registered trademark of NewTek. Copyright 1986 DIGI-PIX.

```
Minus12:
IF mthl-12<1 THEN Year1
mthl=mthl-12:mthend=mthl+11:GOTO DoList
DoChange:
req=3:GOSUB Requestor1:WINDOW 2:GOTO Year1
DoList:
totint=0:totprin=0:LOCATE 6,22:COLOR Blk,Cyn
FOR n=mthl TO mthend
    totint=totint+tabx(n,1):totprin=totprin+tabx(n,2)
    LOCATE ,22:PRINT USING "####.##";n;
    PRINT USING "####.##";tabx(n,1);tabx(n,2);tabx(n,0)
NEXT
IF mthend-mthl<11 THEN
    FOR n=mthend+1 TO mthl+11
        LOCATE ,22:PRINT SPACE$(36)
    NEXT
END IF
LOCATE 18,25:PRINT USING "####.##";totint;totprin
COLOR Blu:LOCATE 18,21:PRINT"Total:"
IF sw=0 THEN
    sw=1
    SAY TRANSLATE$("Click in appropriate box to continue"),voice%
END IF
IntRtnWhat:
boxx=1a:boxy=1b:GOSUB GetGadget
GOTO DoRequest
```

```
IntPrint:
GOSUB POpen:GOSUB NewPage
Pval=20:GOSUB PHorSpc:GOSUB PBldOn
PRINT #Pnum,"INTEREST & PRINCIPAL PAYMENTS BY MONTH"
GOSUB PBldOff:Pval=2:GOSUB PVrtSpc
Pval=28:GOSUB PHorSpc:PRINT #Pnum,"Loan Amount:";
PRINT #Pnum,USING "####.##";
amtPval=26:GOSUB PHorSpc:PRINT #Pnum,"Interest Rate:";
PRINT #Pnum,USING "####.##";itr:PRINT #Pnum,"%";
Pval=24:GOSUB PHorSpc:PRINT #Pnum,"Number of Years:";
PRINT #Pnum,USING "####.##";yrs:PRINT #Pnum," (";
PRINT #Pnum,USING "####.##";mths:PRINT #Pnum," months)"
Pval=24:GOSUB PHorSpc:PRINT #Pnum,"Monthly Payment:";
```

```
PRINT #Pnum,USING "####.##";pay
Pval=2:GOSUB PVrtSpc:Pval=19:GOSUB PHorSpc
PRINT #Pnum,"Month Interest Principal Balance"
totint=0:totprin=0
FOR n=mthl TO mthend
    totint=totint+tabx(n,1):totprin=totprin+tabx(n,2)
    Pval=20:GOSUB PHorSpc:PRINT #Pnum,USING "####.##";n;
    PRINT #Pnum,USING "####.##";tabx(n,1);tabx(n,2);tabx(n,0)
NEXT
Pval=18:GOSUB PHorSpc:PRINT #Pnum,"Total:";
PRINT #Pnum,USING "####.##";totint;
PRINT #Pnum,USING "####.##";totprin
Pval=4:GOSUB PVrtSpc:GOSUB PClose
GOTO IntRtnWhat
```

```
IntEnd:
GOTO Main
```

```
Requestor1:
WINDOW 4,,(40,16)-(280,152),0,1
COLOR Blk,Cyn:CLS
PRINT"Please enter requested values"
PRINT"and press RETURN after each."
IF req=3 THEN GetStrt
LOCATE 6,3:PRINT"(100 to 900000)"
GetAmt:
LOCATE 5,7:PRINT"Loan Amount: "
LINE (150,30)-STEP(50,10),,b:PAINT (155,35),Blu,Blk
LOCATE 5,20:COLOR Blk,Blu:LINE INPUT"",amt$:amt=VAL(amt$)
COLOR Blk,Cyn:IF amt<100 OR amt>900000 THEN GOSUB ReqErr:GOTO
GetAmt
LOCATE 6,3:PRINT SPACE$(15)
LOCATE 8,3:PRINT"(2.00 to 30.00)"
GetInterest:
LOCATE 7,5:PRINT"Interest Rate: "
LINE (150,46)-STEP(42,10),,b:PAINT (155,48),Blu,Blk
LOCATE 7,20:COLOR Blk,Blu:LINE INPUT"",itr$:itr=VAL(itr$)
COLOR Blk,Cyn:IF itr<2 OR itr>30 THEN GOSUB ReqErr:GOTO
GetInterest
LOCATE 8,3:PRINT SPACE$(15)
LOCATE 10,8:PRINT"(1 to";maxyrs;")"
GetYears:
LOCATE 9,6:PRINT"Number Years: "
LINE (150,62)-STEP(18,10),,b:PAINT (155,64),Blu,Blk
LOCATE 9,20:COLOR Blk,Blu:LINE INPUT"",yrs$:yrs=VAL(yrs$)
COLOR Blk,Cyn:IF yrs<0 OR yrs>maxyrs THEN GOSUB ReqErr:GOTO
GetYears
mths=yrs*12
LOCATE 10,8:PRINT SPACE$(10)
GOTO ReqExitGetStrt:LOCATE 6,10:PRINT"(1 to 12)"
LOCATE 5,1:PRINT"Months in 1st Year: "
LINE (158,30)-STEP(18,10),,b:PAINT (160,32),Blu,Blk
LOCATE 5,21:COLOR Blk,Blu:LINE
INPUT"",mths1st$:mths1st=VAL(mths1st$)
COLOR Blk,Cyn:IF mths1st<1 OR mths1st>12 THEN GOSUB
ReqErr:GOTO GetStrt
LOCATE 6,10:PRINT SPACE$(9)
GOTO ReqExit
ReqErr:
SAY TRANSLATE$("Please stick to the indicated range."),voice%
RETURN
ReqExit:
SAY TRANSLATE$("Thank you!"),voice%
boxx=Ra:boxy=Rb:GOSUB DrawGadgets
boxx=Ra:boxy=Rb:GOSUB GetGadget
WINDOW CLOSE 4:RETURN
```

```
NewPage:
IF PCount=0 THEN
    Pval=3:GOSUB PVrtSpc
    Pval=11:GOSUB PHorSpc:GOSUB PBldOn
    PRINT #Pnum,"The Loan Information Program":GOSUB PBldOff
    Pval=29:GOSUB PHorSpc:PRINT #Pnum,"Bryan D. Catley"
    Pval=2:GOSUB PVrtSpc
ELSE
    IF PCount MOD 2=0 THEN
        GOSUB PFFeed:Pval=7:GOSUB PVrtSpc
    END IF
END IF
PCount=PCount+1
RETURN
```

```
QuitRoutine:
IF PCount<>0 THEN
    GOSUB POpen:GOSUB PFFeed:GOSUB PClose
END IF
```

```
voice$(7)=0
SAY TRANSLATE$("O K. Talk to you again soon!"),voice$
CLS:WINDOW CLOSE 2:SCREEN CLOSE 1:END
```

```
BldGadgets:
FOR n=0 TO numbx-1
  FOR m=0 TO 6
    READ bx(n,m)
  NEXT m
  READ bxtxt$(n)
NEXT n
RETURN
```

```
DrawGadgets:
FOR n=boxx TO boxy
  x1=bx(n,0):y1=bx(n,1):x2=x1+bx(n,2):y2=y1+bx(n,3)
  bg=bx(n,4):fg=bx(n,5):bo=bx(n,6)
  LINE(x1,y1)-(x2,y2),bg,bf:LINE(x1,y1)-(x2,y2),fg,b
  LINE(x1+2,y1+2)-(x2-2,y2-2),fg,b
  LINE(x2+1,y1+1)-(x2+1,y2+1),bo
  LINE(x2+1,y2+1)-(x1+1,y2+1),bo
  COLOR fg,bg:row#=INT(y1/8+2):clm#=INT(x1/8+2)
  LOCATE row#,clm#:PRINT bxtxt$(n)
NEXT n
RETURN
```

```
GetGadget:
type=0
WHILE type=0
  WHILE MOUSE(0)=0:WEND
  mx=MOUSE(1):my=MOUSE(2)
  FOR n=boxx TO boxy
    IF mx>bx(n,0) AND mx<bx(n,0)+bx(n,2) THEN
      IF my>bx(n,1) AND my<bx(n,1)+bx(n,3) THEN
        x1=bx(n,0)+2:y1=bx(n,1)+2
        x2=x1+bx(n,2)-4:y2=y1+bx(n,3)-4
        bg=bx(n,4):fg=bx(n,5)
        LINE(x1,y1)-(x2,y2),fg,bf
        COLOR bg,fg:row#=INT(y1/8+2):col#=INT(x1/8+2)
        LOCATE row#,col#:PRINT bxtxt$(n)
        type=n-boxx+1:n=boxy
      END IF
    END IF
  NEXT n
  WHILE MOUSE(0)<>0:WEND
WEND
boxx=type+boxx-1:boxy=boxx:GOSUB DrawGadgets
RETURN
```

```
POpen: 'Open Parallel Port
IF Pnum=0 THEN Pnum=1
OPEN "PAR:" FOR OUTPUT AS #Pnum:GOTO PCExit
PClose: 'Close Parallel Port
CLOSE #Pnum:GOTO PCExit
PFFeed: 'Form Feed to Next Page
PRINT #Pnum,CHR$(12);:GOTO PCExit
PDbblon: 'Double Width On
PRINT #Pnum,CHR$(27);"W";CHR$(1);:GOTO PCExit
PDbloff: 'Double Width Off
PRINT #Pnum,CHR$(27);"W"+CHR$(0);:GOTO PCExit
PBldon: 'Bold On
PRINT #Pnum,CHR$(27)+"E";:GOTO PCExit
PBldoff: 'Bold Off
PRINT #Pnum,CHR$(27)+"F";:GOTO PCExit
PHorSpc: 'Horizontal Space
PRINT #Pnum,CHR$(27)+"f"+CHR$(0)+CHR$(Pval);:GOTO PCExit
PVrtSpc: 'Vertical Space
PRINT #Pnum,CHR$(27)+"f"+CHR$(1)+CHR$(Pval);:GOTO PCExit
PCExit:
RETURN
```

```
SUB Logo80 (Depth%) STATIC
SHARED Blk,Blu,Grn,Cyn,Red,Mag,Yel,Wht
IF First=0 THEN
  First=1
  SCREEN 1,640,200,Depth%,2
  WINDOW 2,,,16,1
  PALETTE 0,0,0,0 :Blk=0:'Black
  PALETTE 1,0,0,1 :Blu=1:'Blue
  PALETTE 2,0,.5,0 :Grn=2:'Green
  PALETTE 3,0,1,1 :Cyn=3:'Cyan
  PALETTE 4,1,0,0 :Red=4:'Red
  PALETTE 5,1,0,1 :Mag=5:'Magenta
  PALETTE 6,1,.75,0:Yel=6:'Yellow
  PALETTE 7,1,1,1 :Wht=7:'White
END IF
```

AMIGA HARD DISK BACKUP HARDHAT

Full/Incremental/Directory/Single File backup to microdisks. Option list allows skipping of files by name with wildcards. Catalog file provides display of backed up files by name with size, location and datestamp. Double data compression reduced disk space. Printer interface. Uses CLI or Workbench. Multitasking provides background operation. — \$69.95

AMIGA DISK FILE ORGANIZER ADFO

Having trouble finding that file somewhere in your stack of floppys? Can't find all the copies of a particular file? ADFO maintains a database of directories and filenames from your collection of disks. Fast response inquiries return location and last update information. Printer interface. Uses CLI or Workbench. 512K ram and 2 drives recommended — \$59.95.

AMIGA SPELLING CHECKER SPEL-IT

Uses 40,000 word primary dictionary and optional second dictionary. Add/Delete words to both dictionaries. Includes plurals. Text wordcount totals. Uses CLI or Workbench, Mouse or keyboard. — \$49.95

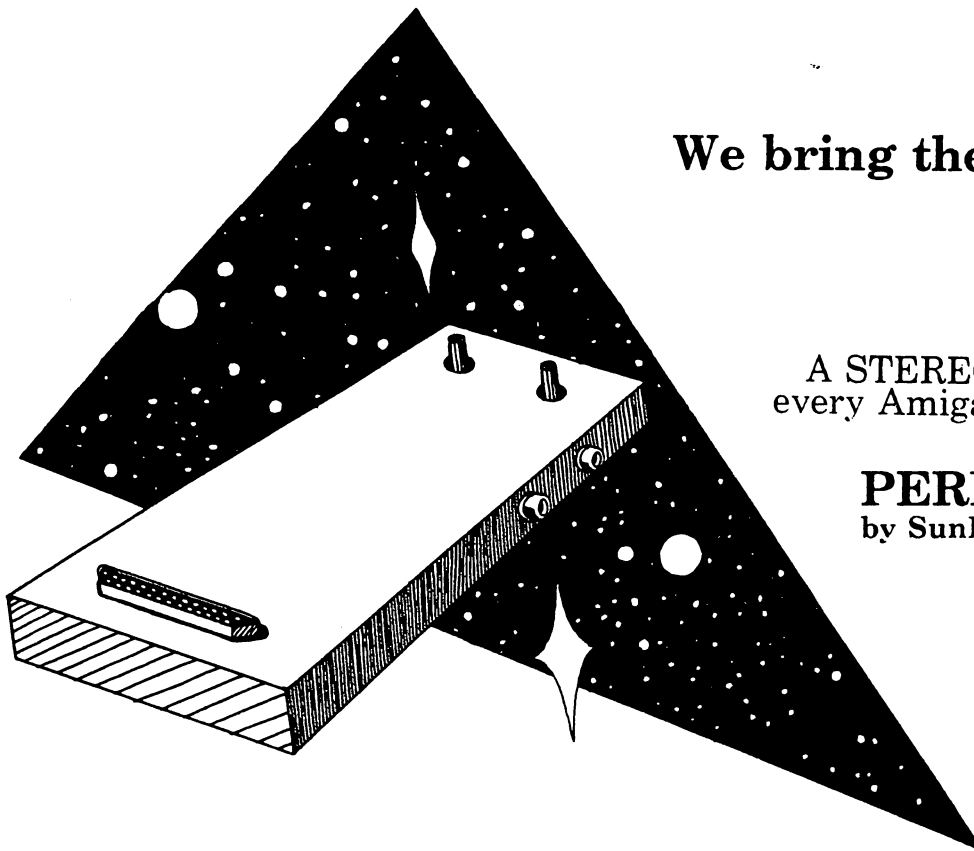
Include \$3.50 S&H Mastercard/Visa Accepted
Calif. Residents Add 6½% Sales Tax

Westcom Industries

3386 Floyd
Los Angeles, CA 90068 (213) 851-4868
Order phone 1 800 621-0849 Ext. 494

```
COLOR ,Blk:CLS
AREA(376,8):AREA STEP(64,0):AREA STEP(-20,16)
AREA STEP(0,24):AREA STEP(-24,0):AREA STEP(0,-24)
COLOR Blu:AREAFILL
AREA(360,8):AREA STEP(32,0):AREA STEP(0,12)
AREA STEP(-16,0):AREA STEP(0,4):AREA STEP(8,0):AREA STEP(0,8)
AREA STEP(-8,0):AREA STEP(0,4):AREA STEP(24,0):AREA STEP(0,12)
AREA STEP(-40,0):COLOR Grn:AREAFILL
AREA(328,8):AREA STEP(24,0):AREA STEP(0,28)
AREA STEP(24,0):AREA STEP(0,12):AREA STEP(-48,0)
COLOR Red:AREAFILL
AREA(272,8):AREA STEP(64,0):AREA STEP(0,12)
AREASTEP(-20,0):AREA STEP(0,28)
AREA STEP(-24,0):AREA STEP(0,-28)
AREA STEP(-20,0):COLOR Cyn:AREAFILL
AREA(264,8):AREA STEP(16,0):AREA STEP(24,40)
AREA STEP(-16,0):AREA STEP(-8,-12)
AREA STEP(-16,0):AREA STEP(-8,12)
AREA STEP(-16,0):COLOR Mag:AREAFILL
AREA(200,8):AREA STEP(56,0):AREA STEP(0,16)
AREA STEP(-24,0):AREA STEP(0,-4)
AREA STEP(-8,0):AREA STEP(0,16)
AREA STEP(8,0):AREA STEP(0,-4)
+AREA STEP(24,0):AREA STEP(0,16)
AREA STEP(-56,0):COLOR Yel:AREAFILL
COLOR Blu,Blk:LOCATE 24,9
PRINT"Bryan D. Catley 1239 Portner Road Alexandria Virginia 22314";
END SUB
```

•AC•



We bring the AMIGA to life...

A STEREO sound digitizer that every Amiga owner should have!

PERFECT SOUND
by SunRize Industries

FUN: RECORD SOUNDS, PLAY THEM BACK FASTER OR SLOWER, EVEN PLAY THEM BACKWARDS

AFFORDABLE: ONLY \$79.95

APPLICATIONS: ADD SPEECH, MUSIC AND SOUND EFFECTS TO YOUR PROGRAMS

SPECIFICATIONS:

- Two channel digitizer that records in stereo
- Includes superb editing software
 - IFF compatible
 - Instrument editing
 - "C" source code
 - Graphs, file compression, and more
- Typeset manual
- Library of recorded sounds
- Free technical support



MICROSEARCH

9896 Southwest Freeway
Houston, Texas 77042

(713) 988-2818

Dealer Inquires Invited

STARTING YOUR OWN AMIGA RELATED BUSINESS

"All the unmet needs of the Amiga users of the world are waiting for someone to provide that product which is too good to pass up."

by William Simpson

The Amiga is a new computer, and following its introduction new hardware and software products are coming to the market place. Right now, the demand for those new products is high, but the larger, entrenched companies are being conservative and cautious about introducing their products until they feel sure that the economic environment is appropriate. In other words, the larger firms with reputations and capital to lose are waiting until they determine that Commodore and Amiga will economically survive.

The larger manufacturers want to be sure that enough Amigas have been sold to insure that there are sufficient users to buy their products. This hesitancy has been consistently exhibited with the introduction of every new computer system.

However, with the introduction of new systems, and the predictable dearth of hardware and software to support the new system; new, smaller entrepreneurs have filled that void and attempted to market their product while the demand was high and the competition minimal.

It is to those of you who recognize such an environment surrounding the Amiga, and are considering doing something about it, that this article is addressed.

It must be understood that information concerning starting a small business must, due to the variety of different laws and regulations in each city, county and state; be of a general nature. If after reading this article your fervor to do business has increased, look for books published in your area concerning the specific laws that apply to your intended business. If, after further research you still intend to go ahead, contact an attorney for direct, specific advice relating to your enterprise.

CHOOSING YOUR ORGANIZATIONAL STRUCTURE

The first major step in starting your own business is a consideration of the form the business will take. Do you want your business to be a sole proprietorship, a partnership, a corporation, or some other type of entity? Each type has its own advantages and disadvantages.

Sole Proprietor

A sole proprietorship is the simplest form of business organization. You are the business and the business is you. The major advantage is that you can own your own business without the paperwork and expense of forming a corporation. In addition, you are able to maintain complete control; rather than sharing it in a partnership or corporate setting.

The major disadvantage of a sole proprietorship is the impossibility of separating your assets from those of the business. The debts of the business are your personal debts. If your business becomes liable because of some event; an injury of a customer, etc.; you personally become liable as well.

The sole proprietorship also exhibits the least flexibility in raising capital. You cannot sell an ownership interest to someone else and your ability to borrow money depends on your personal financial condition.

Partnership

Another available option is the partnership. There are two basic types of partners; general partners and limited partners. A partnership may be made up of any number of either type of partners, however, there must be at least one general partner and one other partner, limited or general.

In a limited partnership, the limited partners have liability only to the extent of their investment. However, a limited partner can lose his limited liability if he or she takes part in the control or decision making responsibilities of the business. Control is maintained by the one or more general partners. Changes in the number of limited partners is not as disruptive as the death or retirement of a general partner.

Whether in the form of a limited or regular partnership; the primary advantage of either type of partnership is its flexibility. Partners can make any type of arrangement they like as to the responsibilities and duties of each partner; so long as the arrangement is not illegal.

There is no requirement, as in a corporation, that ownership interests in capital and profits be equal to the investment made by each. Also, the ability to raise capital is greater than in a sole proprietorship; though not as great as in a corporation.

The primary disadvantage of a partnership is that it is not considered a separate legal entity. That is, the general partners are personally responsible for the debts and liabilities of the business, much as a sole proprietor.

Corporation

The third primary form of business organization is the corporation. A corporation, if formed and maintained correctly, is considered a separate legal entity from those who own or manage it. Most of the characteristics of a corporation stem from its status as a

GREAT COVER-UPS.

Protect your investment
with opaque vinyl covers.



Amiga & Monitor (Stacked)	8.95
Computer-A1000	4.95
Monitor-1080	6.95
Keyboard	2.95
System Pack (3 pcs)	12.95
(Computer-A1000, Monitor-1080, Keyboard)	
3½" or 5¼" Disk Drive	2.95
Printer Covers	4.95
(Specify Make and Model Number)	

Choose from: Panasonic, Okidata, Epson, Toshiba,
Silver Reed, Canon, Fujitsu or Star
Micronics

Please add \$1.00 per item (Max. \$3.00) for
postage and handling.

Send order to: (503) 246-8977

GREAT COVER-UPS
6805 SW 8th Ave.
Portland, OR 97219

separate entity. They include limited liability, perpetual existence, free transferability of ownership shares, and the corporation's ability to own property and bring and defend lawsuits in its own name.

The management of a corporation is divided into three parts; shareholders, directors, and officers. In similar corporations, it is possible for the same individuals to function in all three capacities. However, in order to maintain the corporation's existence separate from those who earn it, it is imperative that the duties and responsibilities of each of the three capacities be performed strictly according to law. If the rules are not strictly followed, the separate existence of the corporation may be destroyed and the shareholders, directors or officers may be held personally responsible for the debts or liabilities of the corporation.

Thus, the most obvious benefit of the corporation status can be lost through failure to run the corporation as a corporation.

Concerning the tax aspects of choosing a corporate form of business organization; it should be kept in mind that the tax laws are undergoing a major revision. Thus, it is essential that you obtain the advice of your tax lawyer or accountant to determine the effects of your prospective business decisions.

Generally, except for Subchapter S corporations, a corporation must pay tax on its net taxable income, and the shareholders must pay their tax on the corporation's net earnings that are distributed in the form of taxable dividends. In essence, double taxation.

The disadvantages of using the corporate structure include the inability to restrict the transfer of shares to outsiders. This is important if you do not want outsiders taking an ownership interest in your business.

Further, the tripartite management (shareholders, directors, and officers) required in order to maintain the corporate status may be too cumbersome for small businessmen.

The double taxation discussed above is obviously a disadvantage, however, when fringe benefits are considered, corporations enjoy distinct tax advantages over partnerships and sole proprietorships under the current tax laws.

Shareholders who are employed by the corporation can qualify as employees and are eligible for special insurance programs and other fringe benefits capable of creating advantageous tax results.

The specific form of corporation that is probably most advantageous to most of us is known as a subchapter S corporation.

The purpose of the S corporation is to maintain the limited liability of a corporation and the tax advantages of other forms of business; thus avoiding the double taxation usually found in a regular corporation.

The basic requirements are that the corporation have only one class of stock and a maximum of 35 shareholders; all of whom are individuals. Clearly, such requirements are typical of smaller, newly formed corporations.

The state governments create the rules concerning the requirements for formation of a corporation. Usually, the steps required include the preparation of Articles of Incorporation, which state the proposed name of the corporation along with a statement of the purposes for which the corporation is formed. Additionally, the Articles usually state the names and addresses of the incorporators, the location of the principal office, the names of the intended first shareholders, the type of capital stock issued and maximum amount intended, and the capital required at the time of incorporation.

In addition to preparation of the Articles of Incorporation, it is necessary to reserve the corporation's name with the state so that no other business can take the name while your corporation is in the process of being formed. Lastly, the minutes of the first meeting of the Board of Directors and the bylaws of the corporation must be created.

Depending on your city and state and the complexity of the documentation required, an attorney might charge from \$500 to \$1500 to create the required documentation and take care of the necessary filings and fees.

There are several excellent books available on how to do your own incorporation. In the event you purchase one of those books, be sure it is an up to date edition; as the law tends to change on nearly a yearly basis.

There are other forms of business organizations available if neither the sole proprietorship, partnership, nor corporation meet your needs. They include limited partnership associations, joint ventures, business trusts, cooperatives and franchises. To obtain more information about these types of companies, consult an expert in your area.

--- OTHER NECESSARY STEPS ---

Business Licenses

After you decide on the appropriate form of business, there are several other steps you will want to consider.

First, business licenses and permits are required by most local governments; primarily the city and county although some businesses require state and even federal licenses.

Fictitious Business Name

If you intend to name your business something other than your real name, you will be required to file a Fictitious Name Statement with the county you intend to do business. Such a filing prevents any other businesses from using your business name.

In addition to the filing statement, you will also be required to publish the Fictitious Name Statement in a newspaper. Probably the county clerk in your area will be able to suggest newspapers that are permitted to publish your statement.

Also, it is necessary to renew your statement on a periodic basis, usually every five years.

Sales Tax

If your state has a sales tax, you will be required to collect the tax from your customers and pay it to the state. Depending on the volume of business that you do, you will be required to prepare tax returns on a monthly, quarterly, semi-annual, or annual basis.

You will also be issued a resale permit giving you the right to purchase goods for resale without paying the sales tax to your supplier. Only if you intend to resell your items as part of your regular business, can they be purchased in this manner. Therefore, it is illegal to purchase items for personal use by this method.

Federal Identification Number

As you prepare your taxes and apply for licenses, you will identify yourself either by using your Social Security Number or by a Federal Employer Identification Number. If your business is a sole proprietorship, the Social Security Number will be sufficient until you start hiring employees.

When you hire your first employee, you will be required to apply for a Federal Employer Identification Number. This is done by handing in form SS-4 to the IRS. Once you have been issued a number you will be required by IRS to prepare payroll tax returns whether you have employees or not. Partnerships and corporations must have the ID number from either the state or the federal government whether they have employees or not.

Interstate Commerce

Any business engaged in interstate commerce must obtain a federal license for that purpose. For more information contact the *Federal Trade Commission, Washington, D.C. 20508.*

Turbo Screendump

by Heart & Mind Software

THE Professional Full-Featured screen printer for the AMIGA that produces graphics printouts from 1 x 1 inches to 10 x 10 FEET with any standard Amiga compatible printer.

- * NOT JUST A SIMPLE SCREEN PRINTER. You control over 30 parameters in picture/text printout OR let TurboScreenDump do it all!
- * COMPLETE control of image capturing and sizing
- * COMPLETE control of printout including sizing, aspect ratio, resolution, and placement
- * Works with graphics images made with Deluxe Paint, Aegis Images, Aegis Draw, Graphicraft, Deluxe Print, and all word processors -- Even AmigaBasic Drawings!
- * Excellent for paste-ups, copy work, custom layouts, and mixing text and graphics

PRICE: \$49.95

Available for IMMEDIATE delivery.
GRONERT COMPUTERS, LTD.
4505 Forest Avenue
Des Moines, Iowa 50311
515-255-3735

Visa MC Check Money Order

COPYRIGHT PROTECTION SOFTWARE

As many of you are probably considering forming a company to market that new program you have created, a brief introduction to the copyright protection of software is also in order.

The current federal statutes state that a copyright extends to works that have been fixed in any tangible means of expression from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device.

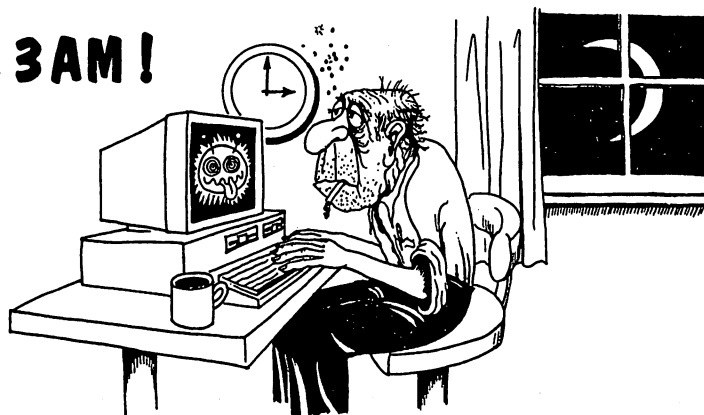
A computer code, whether in the form of object code or source code is a "literary work" as that term is described in the Copyright Program. Therefore, it is protected from unauthorized copying of either its object or source code version. Also, a program that has been "fixed" in a ROM Chip may also be copyrighted.

Patents

Patents, on the other hand, according to the federal statutes, are issued for a new and useful process, machine, manufacture, or composition of matter. The courts have held that the instructions or formula contained in a computer program can be patented if the program is part of an otherwise patentable process, but only as part of that process.

Laws of nature, natural phenomena, and abstract ideas may not be patented. Therefore, an algorithm, which the courts have held as a law of nature, cannot be patented.

It's 3 AM !



Do you know where your bugs are ?

This C programmer is finding his bugs the hard way...one at a time.
That's why it's taking so long. But there's an easier way. Use

Amiga-Lint 2.00

Amiga-Lint analyzes your C programs (one or many modules) and uncovers glitches, bugs, quirks, and inconsistencies. It will catch subtle errors before they catch you. By examining multiple modules, Amiga-Lint enjoys a perspective your compiler does not have.

- NEW: ANSI C extensions (enum, prototypes, void, defined, pragma) and many additional checks.
- Full K&R C
- Use Amiga-Lint to find:
 - inconsistent declarations
 - argument/parameter mismatches
 - uninitialized variables
 - unaccessed variables
 - unreferenced variables
 - suspicious macros
 - indentation irregularities
 - function inconsistencies
 - unusual expressions
 - ... MUCH MUCH MORE
- User-modifiable library-description files for the Aztec and Lattice C compilers.
- All warning and informational messages may be turned off individually.
- Indirect files automate testing.
- Use it to check existing programs, novice programs, programs about to be exported or imported, as a preliminary to compilation, or prior to scaling up to a larger memory model.
- All one pass with an integrated pre-processor so it's very fast.
- Has numerous options and informational messages.
- It will use all the memory available.
- PRICE: \$98.00 MC, VISA, COD (Includes shipping and handling within US) PA residents add 6% sales tax. Outside USA add \$15.00. Educational and quantity discounts available.
- Trademarks: Amiga-Lint(Gimpel Software), Amiga(Commodore)

GIMPEL SOFTWARE

3207 Hogarth Lane • Collegeville, PA 19426
(215) 584-4261

STEPS TO COPYRIGHTING

To obtain the forms necessary to copyright your software, write to: *Information and Publication Section, Copyright Office, Library of Congress, Washington, D.C., 20559.* They will send the forms and more detailed information free of charge.

In most instances, a lawsuit for copyright infringement cannot be started unless the copyright has been registered with the Copyright Office. Therefore, registration is the primary step that must be taken in order to insure protection for your creation.

To register your copyright claim, you must submit an application along with the required registration fee and copies of your work to the Copyright Office. Your claim will then be registered and a certificate of registration will be issued if it is determined that the material you deposited is copyrightable. The current fee is \$10.00, although you should request the current fee schedule when requesting the appropriate forms.

The effective date of copyright registration is the date on which the application, deposit and fee were received in the Copyright Office. After the registration process is complete, your copyright of the work continues for your life plus 50 years. This applies whether your creation is published or not. However, a notice of copyright must be placed on all publicly distributed copies of your copyrighted software. The copyright notice consists of three separate parts:

1. The designation for "copyright" by either:
 - a. The letter "C" contained within a circle;
 - b. The full word "copyright";
 - c. Or the abbreviation "copr."
2. The year of the first publication of the software.
3. The identity of the copyright owner.

Hopefully, this brief survey of the considerations necessary to forming your new business has been helpful and useful in stimulating you to learn more about the complex but rewarding world of business. All the unmet needs of the Amiga users of the world are waiting for someone to provide that product which is too good to pass up. Maybe that product is yours!

•AC•

KEEP TRACK OF YOUR BUSINESS USAGE FOR TAXES

"Uncle Sam may require you to defend your business usage of the computer."

By James W. Kummer

The Amiga Computer can be a very powerful tool for your business. With the programs now and soon to become available, you can prepare presentations, lay-up advertising copy, track your inventory with a spread sheet, and so on. But what about those games? Even on the computer you use for your business, you may be tempted to purchase a game or two, for the spectacular graphics and animation that the Amiga is capable of producing.

When you file your next income tax return, your Uncle Sam may require you to defend your business usage of the computer. The Internal Revenue Service will expect to see a log of the use of your business computer. This should be a record of who used it, for what purpose, and for what period of time.

You say you have already started your log? You keep a notebook, or a calendar pad right beside the computer? Good, then, you are the conscientious computer user who will have no problems with the IRS. You also are probably aware of the pain in the neck that keeping such a written log can be.

Why not let your computer, that work-saving marvel, keep your log for you? Your Amiga automatically executes a startup-sequence (that's the name of the file, by the way) every time you turn it on or re-boot (by simultaneously depressing the 'CTRL' key and both 'Amiga' keys). We will show you how to make this startup-sequence procedure work for you, and to make the entries into a usage log.

Within the directory 's' of your 'workbench' disk, there is the file 'startup-sequence'. This file prompts the user to set the time with the 'preferences' feature. While you can set time with preferences, it is a little cumbersome - and until now, you may have seen no reason to even bother setting the time.

With a minor modification to the 'startup-sequence' file, you don't have to go to 'preferences' to set the time, and further the computer will allow you to enter the name of the user and the purpose of this session of computer usage.

The computer will build a record file for you, recording the name of the user and the business/pleasure usage category of this session of operation. Forget your log sheets, the notebook sitting beside the computer. Just respond to a few simple prompts and procedures, and your purpose of employing the computer will be entered into a database for future reference.

Future reference? What good does the information do you if it sits in your computer, in memory or on disk? Very little, unless you can extract the data, and prepare a summary of the needed information. Fair enough! We'll give you a program that will prepare such a summary.

First for the file 'startup-sequence'. The file 's/startup-sequence' on your workbench disk must be changed, so that it will better serve your needs to keep track of the computer usage statistics. Use an editor, such as 'ED', to modify your 's/startup-sequence' file to look like the following:

```
DATE TO RAM:time1
ECHO "Terminate next two entry-lines with CTRL-\"
ECHO " "
ECHO "Enter 'date', (dd-mmm-yy) and time (hh:mm)"
EXECUTE *
DATE TO RAM:time2
ECHO "Enter name, '--', and usage category"
COPY * TO RAM:time3
COPY SYS:s/usage to RAM:time4
JOIN RAM:time4 RAM:time1 RAM:time3 RAM:time2 AS
SYS:s/usage
DELETE RAM:time? QUIET
ECHO " "
LoadWb
; ENDCLI > NIL:
```

This is your 'startup-sequence' file. Under its control, your Amiga prompts you to enter 'date', including day, month, year, hour and minute. Note that you must enter the word 'date' before entering date and time, and terminate the entry with a 'CTRL-\''. The month entry is the first three letters of the month. Day, month and year are separated dashes, and a colon separates hour (24-hour clock time) from minute.

Next you are asked to enter the user name (nickname, initials, or whatever you might choose to employ to identify your users) and a characterization of the purpose of this session of usage. Terminate this entry line likewise with a 'CTRL-\''. The name of the file that 'startup-sequence' stores this information in is (surprise!) 'usage' within the 's' directory. Note that, on the line containing name and purpose, a double dash must be supplied to separate these two fields.

Remember, when providing the 'CTRL-\' sequence, you first depress the 'CTRL' key, hold it down, depress the ']' and release both of them immediately. Don't worry if you accidentally hit the 'return' key instead of the 'CTRL-\''. It is such a force of habit, to

poke that 'return' key when you reach the end of a line or input sequence. If you do, you haven't hurt anything at all. Until you hit the 'CTRL-\' key sequence, your computer will just sit there and stare at you. The worst result of hitting the 'return' key is that you may get an unwanted blank line in your 'usage' file.

And speaking of the 'usage' file, the first time you employ the 'startup-sequence' above, you will get an error message, because the file 'usage' does not yet exist. But in the process, the file 'usage' does get created, as an empty file. When you get this error message, your CLI prompt appears, just reboot and repeat the sequence. Now your usage file will be present and updated properly.

Note that the last line of the listed 's/startup-sequence' file has a ';' in the first column. If you delete the semicolon, then, on completion of the startup-sequence, your CLI window will be terminated, and you will see nothing but the workbench options. If you have no use for the CLI window, then you may wish to delete the semicolon to bring you straight to the workbench environment.

After completion of its logging operation, you are left in a CLI. If that's where you want to be, great! Otherwise, there are at least two ways you can get to where you want to be. First, you could type 'ENDCLI'. This will kill your CLI window and take you to the workbench. Or, you could just grab the bottom of the CLI window, shrink it up toward the top of the screen, and then move the shrunken CLI window down to the bottom of the screen. Now, you can select any of the workbench options, and later come back to the CLI window.

Why would you want to come back to the CLI window? Well, for one reason, to terminate this usage session. The system has to be told that this session is over, so that the end-time can be entered into the 's/usage' file. To do this, employ the 'logout' command file. Using the 'ED' editor, enter text into the file named 'logout' on your df0:workbench as follows:

```
DATE TO RAM:time1
COPY SYS:s/usage to RAM:time4
JOIN RAM:time4 RAM:time1 AS SYS:s/usage
DELETE RAM:time? QUIET
ENDCLI > NIL:
```

The manner in which you invoke this file is, in a CLI window, type 'execute logout'. What 'logout' does is to enter another record of 'date/time' into the 's/usage' file. This tabulates the termination of this session of previously stated purpose. It allows you to turn off the computer, or to begin another session for a different user, for a different purpose.

The following is a sample listing of the file 's/usage':

```
Sunday 2-Feb-8617:56:20
jim -- prog-dev
Sunday 2-Feb-8617:56:0
Sunday 2-Feb-8618:8:35
Sunday 2-Feb-8618:9:43
```

```
BILL -- prog-dev
Sunday 2-Feb-8618:11:0
Sunday 2-Feb-8618:14:42
Sunday 2-Feb-8618:16:5
Sunday 2-Feb-8618:16:20
jim -- prog-dev
Sunday 2-Feb-8618:16:34
Sunday 2-Feb-8618:18:22
Sunday 2-Feb-8618:19:42
Sunday 2-Feb-8618:21:53
jim -- games
Sunday 2-Feb-8618:22:11
Sunday 2-Feb-8618:40:59
Sunday 2-Feb-8618:40:55
jim -- prog-dev
Monday 3-Feb-8620:8:0
Monday 3-Feb-8620:51:43
jim -- prog-dev
Tuesday 4-Feb-8620:8:0
Tuesday 4-Feb-8620:48:55
Tuesday 4-Feb-8620:49:15
jim -- prog-dev
Sunday 2-Feb-8613:31:0
Sunday 2-Feb-8615:14:27
Sunday 2-Feb-8615:14:47
tom -- prog-dev
Tuesday 11-Feb-8605:6:0
Tuesday 11-Feb-8605:16:6
Tuesday 11-Feb-8605:16:33
BILL -- games
Tuesday 11-Feb-8606:10:0
Tuesday 11-Feb-8606:14:6
Tuesday 11-Feb-8606:14:27
jim -- prog-dev
Friday 14-Feb-8608:32:0
Friday 14-Feb-8616:7:44
Friday 14-Feb-8616:8:4
jim -- prog-dev
Friday 14-Feb-8619:19:0
Friday 14-Feb-8619:27:13
Friday 14-Feb-8619:27:34
tom -- games
Saturday 15-Feb-8615:9:0
Saturday 15-Feb-8615:50:46
Saturday 15-Feb-8615:51:9
jim -- prog-dev
Sunday 16-Feb-8616:6:0
Sunday 16-Feb-8616:57:34
jim -- prog-dev
Wednesday 19-Feb-8620:4:0
Wednesday 19-Feb-8621:28:18
Wednesday 19-Feb-8621:28:40
jim -- prog-dev
Friday 21-Feb-8617:31:0
Friday 21-Feb-8620:13:2
Friday 21-Feb-8620:13:24
jim -- prog-dev
Saturday 22-Feb-8616:49:0
Saturday 22-Feb-8617:52:31
Saturday 22-Feb-8617:52:52
jim -- prog-dev
Sunday 23-Feb-8609:33:0
Sunday 23-Feb-8611:23:26
Sunday 23-Feb-8611:23:47
jim -- prog-dev
Sunday 23-Feb-8613:55:0
```

Note that the time appears several times between user/purpose lines. This is intentional, to accommodate the eventuality that you

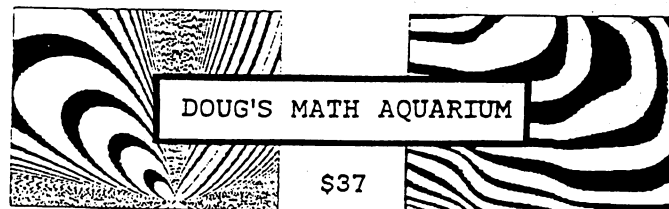
or the other users of your Amiga may forget to execute the 'logout' function at the end of their usage session. Even if they do, a time is entered into the 'usage' file the next time the computer is turned on, and that time is approximately the time that it was last turned off.

Why, then, should you use the 'logout' feature? First, to be sure that your time-of-completion is entered - you may finish your session, but not turn off the computer right away. Second, you may wish to immediately start another session, for a different usage purpose, or for a different user. To do this, you can either 're-boot' with the 'CTRL' and both Amiga keys, or you can simply type in from a CLI window the command 'execute s/startup-sequence'. Either way, the new user/purpose will be entered into your 's/usage' file.

Now, when it comes time to produce a tabular log of usage for the IRS, you can use the following program, in AmigaBASIC, to produce a readable summary:

```
-- USAGE TABULATION PROGRAM LISTING --
'
' -- by James W. Rummer --
NUMU=20 : NUMP=18
DIM
USER$(NUMU), PURPOSE$(NUMP), STATS(NUMU, NUMP), SUMP(NUMP), SUMU(NUMU)
DIM SHARED MO$(12)
GAP$="--"

OPEN "I", #1, "DFO:S/USAGE"
LPRINT "USAGE TABULATION AS OF "; DATE$
LPRINT
FOR I=1 TO NUMU : USER$(I)=""
SUMU(I)=0 : FOR J=1 TO NUMP : STATS(I, J)=0 : NEXT J : NEXT I
FOR J=1 TO NUMP : SUMP=0 : PURPOSE$="" : NEXT J
DATA JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC
FOR I=1 TO 12 : READ MO$(I) : NEXT I
IUSER=0 : IPURP=0 : SUMTOT=0 : Z$="" / "
LOOP:
  INPUT #1, A$
  IF EOF(1) THEN CEOF
  LA=LEN(A$)
  IF LA=0 THEN LOOP
  LDASH=INSTR(A$, "---")
  IF LDASH=0 THEN LOOP
  LUSER=LDASH-1
  NUSER$=LEFT$(A$, LUSER)
  CALL UNBLANK(NUSER$, LUSER)
  NUSER$=UCASE$(NUSER$)
  LPURP=LA-LDASH-1
  NPURP$=RIGHT$(A$, LPURP)
  CALL UNBLANK(NPURP$, LPURP)
  NPURP$=UCASE$(NPURP$)
  FOR I=1 TO IUSER
    IU=I : IF USER$(I)=NUSER$ THEN MATCHP
  NEXT I
  IUSER=IUSER+1 : IU=IUSER
  USER$(IU)=NUSER$
  MATCHP:
  FOR I=1 TO IPURP
    IP=I : IF PURPOSE$(I)=NPURP$ THEN STIME
  NEXT I
  IPURP=IPURP+1 : IP=IPURP
  PURPOSE$(IP)=NPURP$
  STIME:
  INPUT #1, B$ : IF EOF(1) THEN CEOF
  LA=LEN(B$) : IF LA=0 THEN STIME
  IF B$=SPACE$(LA) THEN STIME
  CALL UNDATE(B$, DAY1, HOUR1, LA)
  ETIME:
  INPUT #1, C$ : IF EOF(1) THEN CEOF
  LA=LEN(C$) : IF LA=0 THEN ETIME
  IF BC$=SPACE$(LA) THEN ETIME
  CALL UNDATE(C$, DAY2, HOUR2, LA)
  DDAY=DAY2-DAY1
```



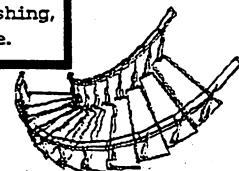
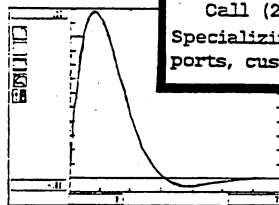
- * Dazzling designs grow on your screen.
- * Teaches math: Makes functions visual.
- * Edit pictures with your paint program.
- * Color Displays.* Wallpaper for the Mind.
- * Simple function entry.* 3-D Displays.
- * Color cycling for apparent motion.
- * Fast expression compiler.

3-D GRAPHICS LIBRARY.

- * 3-D rotations, reflections, translations, zooms, coordinate changes.* Functions to calculate intersections, orthogonals, lengths and angles.* Uses Fast Floating Point Library.
- * Can be called from assembler or C or BASIC.

\$ Call

Distributed by
Seven Seas Software
P.O.B. 411
Port Townsend WA.
98368
Call (206) 385-3771
Specializing in publishing,
ports, custom software.



```
IF DDAY<-26 THEN DDAY=1      '** month roll-over presumed
DTIME=24*DDAY+HOUR2-HOUR1
IF DTIME<0 OR DTIME>24 THEN DTIME=1
**IF presumed logging error THEN set usage interval to 1 hr
STATS(IU, IP)=STATS(IU, IP)+DTIME
SUMP(IP)=SUMP(IP)+DTIME
SUMU(IU)=SUMU(IU)+DTIME
SUMTOT=SUMTOT+DTIME
LPRINT NUSER$, Z$, NPURP$, Z$, B$, Z$, C$, Z$, DTIME
GOTO LOOP
CEOF:
CLOSE #1
LPRINT : LPRINT GAP$ : LPRINT
FOR J=1 TO IPURP
  LPRINT "USER SUMMARY FOR PURPOSE: "; PURPOSE$(J)
  FOR I=1 TO IUSER
    LPRINT USER$(I); Z$; STATS(I, J); Z$; STATS(I, J) / SUMP(J)
  NEXT I
  LPRINT : LPRINT
NEXT J
LPRINT : LPRINT GAP$ : LPRINT
FOR I=1 TO IUSER
  LPRINT "PURPOSE SUMMARY FOR USER: "; USER$(I)
  FOR J=1 TO IPURP
    LPRINT PURPOSE$(J); Z$; STATS(I, J); Z$; STATS(I, J) / SUMU(I)
  NEXT J
  LPRINT : LPRINT
NEXT I
LPRINT : LPRINT GAP$ : LPRINT
LPRINT "GRAND TALLY USAGE STATISTICS"
LPRINT
FOR I=1 TO IUSER
  LPRINT USER$(I); Z$; SUMU(I); Z$; SUMU(I) / SUMTOT
NEXT I
LPRINT : LPRINT
FOR J=1 TO IPURP
  LPRINT PURPOSE$(J); Z$; SUMP(J); Z$; SUMP(J) / SUMTOT
NEXT J
END
```



```

SUB UNBLANK(B$,LB) STATIC
LSIDE:
IF LEFT$(B$,1) <> " " THEN RSIDE
LB=LB-1 : B$=RIGHT$(B$,LB) : GOTO LSIDE
RSIDE:
IF RIGHT$(B$,1) <> " " THEN UEND
LB=LB-1 : B$=LEFT$(B$,LB) : GOTO RSIDE
UEND:
EXIT SUB
END SUB
'-----
SUB UNDATE(T$,DAY,HOUR,LT) STATIC
I1=INSTR(T$," ") : LT=LT-I1+1 : T$=RIGHT$(T$,LT)
I1=INSTR(T$,"-")
DAY=VAL(MID$(T$,I1-2,2))
MON$=MID$(T$,I1+1,3)
FOR I=1 TO 12 : MONTH=I : IF MON$=MO$(I) THEN MFOUND: NEXT I
MFOUND:
DAY=DAY+MONTH
I1=INSTR(T$,":")
HOUR=VAL(MID$(T$,I1-2,2))
I2=INSTR(I1+1,T$,"")
I=I2-I1-1
MIN=VAL(MID$(T$,I1+1,1))
SEC=VAL(MID$(T$,I2+1,2))
HOUR=HOUR+(SEC/60+MIN)/60
EXIT SUB
END SUB

```

```

USER SUMMARY FOR PURPOSE: PROG-DEV
JIM / 18.97111 / .9880214
BILL / .0616684 / 3.21171E-03
TOM / .1683335 / 8.766865E-03

USER SUMMARY FOR PURPOSE: GAMES
JIM / .3133335 / .2907215
BILL / 6.833363E-02 / 6.340227E-02
TOM / .6961117 / .6458762

```

```

PURPOSE SUMMARY FOR USER: JIM
PROG-DEV / 18.97111 / .9837521
GAMES / .3133335 / .016248

```

```

PURPOSE SUMMARY FOR USER: BILL
PROG-DEV / .0616684 / .4743649
GAMES / 6.833363E-02 / .5256351

```

```

PURPOSE SUMMARY FOR USER: TOM
PROG-DEV / .1683335 / .1947301
GAMES / .6961117 / .8052699

```

GRAND TALLY USAGE STATISTICS

```

JIM / 19.28444 / .9509614
BILL / .130002 / 6.410708E-03
TOM / .8644452 / 4.262784E-02

PROG-DEV / 19.20111 / .9468522
GAMES / 1.077779 / 5.314783E-02

```

The above program, written in AmigaBASIC, converts all input user names and purposes to uppercase. Thus your computer users do not have to concern themselves whether the information that they enter is upper or lower case. All user names and purposes will be printed in all upper case. The program assumes that each session will not last longer than 24 hours - a reasonable assumption! After correcting for crossing over month and year boundaries, if the time-difference is still negative, or still exceeds 24 hours, then the program assumes that some error has occurred, such as a power failure wiping out your time in the middle of a session. To handle that eventuality, the program assumes in those cases a simple '1-hour' of usage.

The output of the usage summary program begins with a restatement of the usage history - one line for each session. Next, it presents a summary of usage by purpose, then by user. Finally, the "grand tally" usage statistics are presented, for each user, and for each purpose.

In the three summary tabulations, the next-to-last number is the hours of use, and the last number is the fractional portion for this individual/purpose. The following is a sample output of the program. It was produced from the sample 's/usage' input file shown above. Note that, in the grand tally results, of all the users, JIM was the user 95.1% of the time. And, of all the purposes, PROG-DEV was the selected purpose 94.7% of the time.

USAGE TABULATION AS OF 02-24-1986

```

JIM / PROG-DEV / 2-Feb-8617:56:0 / 2-Feb-8618:8:35 / .2097225
BILL / PROG-DEV / 2-Feb-8618:11:0 / 2-Feb-8618:14:42 / .0616684
JIM / PROG-DEV / 2-Feb-8618:16:34 / 2-Feb-8618:18:22 / 2.999878E-02
JIM / GAMES / 2-Feb-8618:22:11 / 2-Feb-8618:40:59 / .3133335
JIM / PROG-DEV / 3-Feb-8620:8:0 / 3-Feb-8620:51:43 / .728611
JIM / PROG-DEV / 4-Feb-8620:8:0 / 4-Feb-8620:48:55 / .6819439
JIM / PROG-DEV / 2-Feb-8613:31:0 / 2-Feb-8615:14:27 / 1.724167
TOM / PROG-DEV / 11-Feb-8605:6:0 / 11-Feb-8605:16:6 / .1683335
BILL / GAMES / 11-Feb-8606:10:0 / 11-Feb-8606:14:6 / 6.833363E-02
JIM / PROG-DEV / 14-Feb-8608:32:0 / 14-Feb-8616:7:44 / 7.595555
JIM / PROG-DEV / 14-Feb-8619:19:0 / 14-Feb-8619:27:13 / .1369438

TOM / GAMES / 15-Feb-8615:9:0 / 15-Feb-8615:50:46 / .6961117
JIM / PROG-DEV / 16-Feb-8616:6:0 / 16-Feb-8616:37:34 / .8594437
JIM / PROG-DEV / 19-Feb-8620:4:0 / 19-Feb-8621:28:18 / 1.404999
JIM / PROG-DEV / 21-Feb-8617:31:0 / 21-Feb-8620:13:2 / 2.700556
JIM / PROG-DEV / 22-Feb-8616:49:0 / 22-Feb-8617:52:31 / 1.058611
JIM / PROG-DEV / 23-Feb-8609:33:0 / 23-Feb-8611:23:26 / 1.840555

```

The program is sized for a limit of 20 users and 18 purposes. If you need more than that, you can change the NUMU and NUMP values in the program. Additionally, if you need to accommodate more than 20 users on your computer, you may need to consider a large mainframe!

In order to avoid a loss of your records in case a disk should fail, you need to periodically back-up your usage file. Copy it to another floppy. And, whether you collect your usage statistics annually or quarterly, you need to delete your usage file so that the usage statistics can begin to accumulate again. Keep a copy, just in case, so that you can regenerate the summary statistics.

If, in the unfortunate instance that you should lose your tabulation, or your usage file, then you may need to change the name of the file in the "OPEN "I" #1 ..." statement to whatever name you have given to the copy of the usage log.

Bear in mind that using a computer to record your usage statistics is by no means a fool-proof way to collect these data. Anyone can cheat the system, and it doesn't matter if the computer or a 3-ring spiral notebook is the vehicle of the untruth. But the scheme presented in this article makes it harder to forget to log your usage, and it does discourage individuals who want to defeat or take advantage of the system.

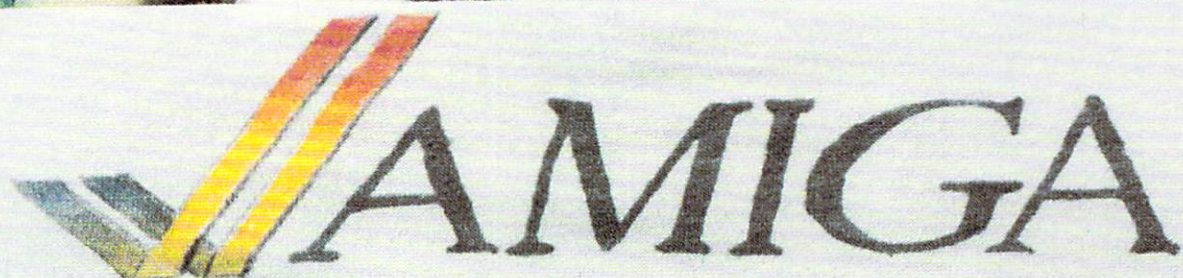
•AC•

Amazing Computing™

Extra!

U.S.A.	\$2.00
Canada	\$2.95

Commodore Amiga™ Information and Programs



Amazing Directory

*Sources and Resources for your
Commodore Amiga™*

Micro-Systems Software —Bigger and Better *for the Amiga*

DATABASE
SPREADSHEET
WORD PROCESSOR

NEW - ORGANIZE! DATABASE MANAGER, VERSION 1.0

Mailing lists! Club memberships! Patient records! Client files! Video tape libraries! Phone call logs! Nearly anything that needs to be filed, sorted or calculated is a candidate for **Organize!**

In seconds, **Organize!** can scan your files, locate information, and display or print in the format you want. Use it to print form letters with the Mailmerge function of Scribble!. Or calculate fields and do statistical analyses of your files with many of the same built-in math functions from Analyze!.

Easily design input forms and output reports with the mouse and pull-down menus. Just as simply - store, sort, review and print. The file size is limited only by disk space and the format is compatible with the industry standard dBASE format.

End your paper shuffle! Get **Organize!** today.

Only \$99.⁹⁵

UPGRADED -

ANALYZE! SPREADSHEET, VERSION 2.0

ANALYZE! FEATURES:

- Pulldown menu interface (mouse-driven).
- Large spreadsheets with efficient memory usage.
- Dedicated function keys for common commands.

NEW ADDITIONS:

- **Business Graphics;** print bar, stacked bar, pie graphs in 2 or 3-D; line, X-Y, area graphs; all in 4 or 8 colors; data from spreadsheets; IFF format; view up to 4 graphs at same time; instantly redraw graphs when data changes; ranges, labels, titles, legends, rotation, scaling; fast and effective!
- **Command Macros;** save keystrokes; create templates.
- **Sorting;** rearrange row or column data quickly.
- **File Icons;** access spreadsheets via icons or names.

NEW PRICE Only \$149.⁹⁵

IMPROVED - SCRIBBLE!

WORD PROCESSOR, VERSION 1.0

SCRIBBLE! FEATURES:

- Pulldown menu interface (mouse-driven)
- Multiple windows; edit/cut & paste 4 documents on screen.
- Preview; see final form on screen before printing.

NEW ADDITIONS:

- **Spellcheck;** expandable 40,000 word dictionary; check word, all words on screen, or entire document; alternative spellings shown.
- **Mailmerge;** print form letters, mailing labels; create data file with Scribble! or Organize!
- **File Icons;** access documents via icons or names; copy documents by pulling icons across workbench.
- Expanded Memory Support; for larger documents.
- More Amiga Keys; menu commands from keyboard or mouse.
- More Flexibility; Wordstar™ commands; scrolling while cut/paste; improved file operations.

Still only \$99.⁹⁵

See your local dealer or call:

**Brown-Wagh
Publishing**

1-800-451-0900

1-408-395-3838 (in California)

16795 Lark Ave., Suite 210, Los Gatos, CA 95030

Amazing Computing's Amazing Directory of Amiga Sources and Resources

Instant Music	Reviewed by Steve Pietrowicz	45
MindWalker	Reviewed by Richard Knepper	47
The Alegra Memory Board	Reviewed by Rick Wirth	49
TxEd	Reviewed by Jan and Cliff Kent	51
Amazing Directory A guide to the sources and resources		53
Amiga Developers A listing of Suppliers and Developers		66
Public Domain		
A condensed listing of Amicus and Fred Fish PDS Disks		69

Read/Write MS-DOS Disks on your Amiga



DOS-2-DOS Transfers MS-DOS Files To and From AmigaDOS!

- Supports single and double sided 5.25" as well as 3.5" 720KB diskettes.
- Converts ASCII file line-ending characters and provides Wordstar compatibility.
- Supports full directory path names, with wild cards in file names.
- Allows selection of MS-DOS and AmigaDOS subdirectory and displays sorted directory listing.
- Formats 3.5" and 5.25" MS-DOS diskettes.
- Provides duplicate file name detection with query/replace options.
- Provides TYPE and DELETE commands.
- Permits renaming of files where file name restrictions occur.
- Remains resident to permit AmigaDOS disk swapping.

Requires standard Amiga with either an external 5.25" or 3.5" disk drive.
This product is available for immediate shipment. Only \$55.00 plus
\$3.00 shipping and handling. CA residents add sales tax. Telephone orders
welcome. Dealer inquiries invited.

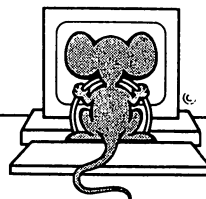


Central Coast Software™

268 Bowie Drive, Los Osos, CA 93402 • 805 / 528-4906

Trademarks: Amiga, AmigaDOS, Transformer, Commodore-Amiga, Inc.; PC-DOS, IBM; MS-DOS, Microsoft; DOS-2-DOS, Central Coast Software.

Mouse Driven



Classic games software you can drive with your mouse! But, you don't need a license -just an AMIGA and:

™ Games Gallery I, II, and III.

Each of these packages contain exciting:
Space, Gambling, Sports Games, and Mind Teasers.

Each provides a standard series of features and options for:

- Speech •Graphics •Menus
 - Color •Help •Voice and •Mouse Control!
- Kickstart 1.1 & 512K memory required. \$29.95
+ \$3.00 shipping & handling.

(713) 488-2144

Telephone orders
welcome

Visa Mastercard Amex



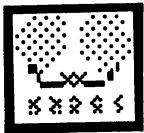
P.O. Box 890408
Houston, TX. 77289-0408

AMIGA is a trademark of Commodore-Amiga, Inc.

SoundScape...Power Play for the AMIGA.



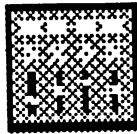
Pro MIDI Studio



The most powerful performance and recording software on any computer. The recording

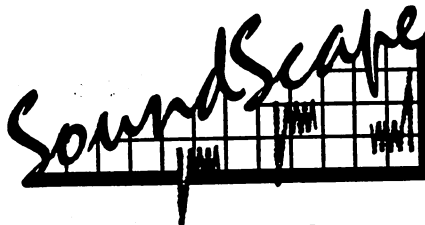
studio-like environment provides complete facilities for routing, recording, editing, transposition and playback of any musical performance. As new modules are introduced, you can "install" them at any time. Music can be performed by the internal sampled sound synthesizer, or with any external MIDI equipment. Record from the QWERTY keyboard or any external MIDI source, including keyboards, guitar and pitch followers. Synchronize with, or provide MIDI clock information, including MIDI Song Pointers. The complete flexibility of the system makes your imagination the only limit to its power.

- Number of notes and tracks determined by available memory
- MIDI patch panel links program modules
- Install new modules at any time
- Up to 16 internal instruments at one time
- Complete sample system with editing, looping, ADSR envelopes, velocity sensitivity, and pitchbend.

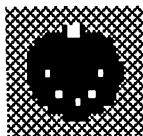


- Up to 160 sampled sounds at one time
- Save and load IFF note and sample files

- Quantize to any multiple of MIDI clock beats
- "Match" mode eases learning of a song
- Complete MIDI sequence and song editing
- Route, merge, split, or bounce any track to any other.



MIDI Interface



Necessary for any program which supports MIDI to communicate with MIDI equipment.

- Completely compatible with the standard Amiga MIDI interface
- MIDI In, Out, and Thru connectors
- Plugs into the serial port



Sound Digitizer



With the SoundScape Sound Digitizer, any sound may be sampled and modified by the Amiga, including voice. IFF File compatibility enables these samples to be used as musical instruments, sound effects, or speech with any IFF compatible music or animation system.

- High quality
- Highest possible fidelity from the Amiga
- Stereo or mono
- Variable sample rates
- Mike and line inputs
- Digitally controlled volume on each channel
- IFF Sample File compatible
- Software included for sampling, editing, and MIDI performance functions

Available From Your AMIGA Dealer.

SoundScape Pro MIDI Studio	\$149.00
AMIGA MIDI Interface	\$ 49.00
SoundScape Audio Digitizer	\$ 99.00

mimetics™

corporation ...the professional software source!!

P.O. Box 60238 Sta. A, Palo Alto, CA 94306 (408) 741-0117

Amiga is a trade mark of Commodore Business Machine

Prices and availability subject to change without notice

Instant Music

..From a Non-Musician's Point of View

"Can this program really let me play music that sounds as if I know what I'm doing?"

Reviwed by
Stephen Pietrowicz

I'm the first person to admit that my musical ability is about zero. I played a couple of different instruments when I was younger, and have forgotten how to play any kind of music at all. When I was considering getting an Amiga, I knew that it would help me regain some kind of musical ability, even if it was just showing off stereo music programs. I was a little more than interested when I saw Instant Music by Electronic Arts. Can this program really let me play music that sounds as if I know what I'm doing?

Instant Music allows for four different instruments to play at a time. When the program starts, it is in "mouse jam" mode. "Mouse Jamming" lets you control one of the instruments that is playing by moving the mouse. The instrument that you control follows the original path that the instrument was intended to, but lets you change the tone of the instrument. By pressing the left mouse button and moving the mouse up and down, you can really make it sound like you know what you're doing.

There are a large variety of songs that are included in the package. Categories include Jazz/Blues, Folk, Classical, and Rock. There are so many songs that you can mouse jam the day away, but what about creating your own songs?

To assist you in creating your own music are three libraries of instruments that are included on the disk. Instruments include drums, piano, harmonica, guitar, and electric bass. You can also use instruments that are not included on the Instant Music disk. (Deluxe Video has instruments on it that aren't found in Instant Music).

Harv Laser (CBM*HARV, the chairman of People Link's Commodore Club) has created a library of new instruments for Instant Music. Dan James, a.k.a DJJAMES, a very active member of PeopleLink, wrote a conversion program that takes the sounds on the disk that many dealers had in the early days of the Amiga and converts them to Instant Music format.

Laser converted all of the sounds, and put them into two large archives. Alarms, pigs oinking, horse whinnies, and a variety of strange noises can now be included into your music. I'm not too sure how I can incorporate sounds like that into a song, but it is a very interesting effect. The archives containing these sounds can be found on People Link in section 13 of Commodore Club.



The manual included with the software was a bit confusing. It started off easily enough, with examples of how to load and save songs, and quickly progressed to techniques used in creating your own songs. Copying, erasing, and pasting notes is very straight forward. The rhythm guides, bass line, chords, and melody sections left me a bit confused. Not having an extensive background in music, how am I supposed to know how to make the song sound the way I want it to sound?

As I said, the editing features of the program are quite good. That will make it much easier for me to try to program my own music. The "QuickDraw" mode of the program makes it easy to enter large amounts of notes, and by using the "Scale Ruler", you can even transcribe music. Transcribing music is more difficult in Instant Music than in other music programs since a musical staff is not displayed to help you place notes.

As I experiment more and more with the program, laying down bass lines, changing the songs supplied with it, and trying to transcribe music, I'm getting more used to it. Instant Music does require that you spend a lot of time with it if you're a beginner trying to create new songs, but at the same time lets you have a lot of fun trying.

•AC•

FASTER THAN CLI!

MORE POWERFUL THAN WORKBENCH!

**ABLE TO LEAP TALL TASKS
IN A SINGLE CLICK!**

ZING! is a utility software package that gives **YOU** the power to access your **AMIGA!** You no longer have to resort to typing cryptic commands through CLI. **ZING!** uses Intuition which provides you with easy window, icon, menu, and mouse controlled features.

Start flying through your system while copying, editing, deleting, renaming, sorting, searching and organizing files and programs. You can save screens to standard IFF files or the printer, monitor and control running tasks, and interface with other software applications.



ZING!™

Of course, **ZING!** has many other powers including a print spooler and a built in screen saver.

ZING! offers these and hundreds of other capabilities without preventing you from running other applications simultaneously. **ZING!** uses Intuition the way it should be used!

Order **ZING!** and transform your mild mannered CLI into the fastest and most powerful computer interface ever conceived! It's available now for the **special introductory price** of

\$79.95 plus \$3.00
for shipping and handling.



P.O. Box 890408
Houston, TX 77289-0408

(713) 488-2144
Credit cards and
dealer inquiries welcome.

AMIGA is a registered trademark of Commodore-AMIGA, Inc.
Workbench and Intuition are trademarks of Commodore-AMIGA, Inc.
ZING! is a trademark of Meridian Software, Inc.

MIND WALKER

"Your objective is to take an introspective journey and piece together the fragments of your personality, thus regaining your sanity."

**Reviewed by
Richard Knepper**

Mind Walker, a game from Commodore-Amiga and Synapse Software, represents the best of the second generation software now becoming available for the Amiga. Mind Walker is not just a port of a program written for another computer, but rather one that was written to take advantage of the power of the Amiga. The result is a program that has an "Amiga feel" to it.

Mind Walker is an arcade and strategy game. Mastering it requires not only quick reflexes, but also good planning. This combination insures that MindWalker will provide entertainment long after the arcade portion of the game has been mastered.

The Game Play

The premise of Mind Walker is nearly as complex as the game itself. You are a physics professor, and after years of study of minute particles and obscure formulae you seem to have gone insane. Your ego has split into four personalities; a human, a wizard, a spriggen (a spider-like humanoid), and a nymph. Your objective is to take an introspective journey and piece together the fragments of your personality, thus regaining your sanity. Sounds easy enough - doesn't it?

The method by which you will accomplish this madness yields the actual game implementation. First, you must unlock a pathway to your brain by tracing a path of coherent thought through your mind. Once in the brain, you must collect the shards of your sanity. Finally, you must piece together these shards in your subconscious and regain your sanity.

Unlocking a passage to your brain is accomplished by tracing a path of coherent thought between a square of crystallized thought and the destination square. In order to keep things interesting there are a number of obstacles that must be overcome.

First, the terrain is jumbled, and the different terrain types require different personalities in order to be able to trace the path. Also, bad thoughts wander about your mind and try to zap you. But never fear, you are armed with your trusty fractal ray. Zap them before they zap you and you won't die.

Once you have successfully traced a path of coherent thought you must then travel to the brain to regain the shards of your sanity. You enter a tube and go through one of the green doors floating by. Inside the brain you find yourself in a maze of neurons. You must travel through this maze, being careful not to bump into the neurons, collect the shards, and make it back to the

door. The problem is that there are nasty viruses floating about that are bent on your destruction. Your defense against them is the thought that you created in your mind. You can use your handy thought reflector to deflect the thought into the viruses, zapping them.

After you have collected the shards of your sanity and returned from behind the green door (don't blame me, I just report this), you find yourself in the subconscious. Here you must piece together the shards of your sanity to form an inkblot. It is here that you find out how cruel the game designers were. You only have seven pieces, and forty-two are needed to complete the puzzle - you must repeat the cycle five more times!

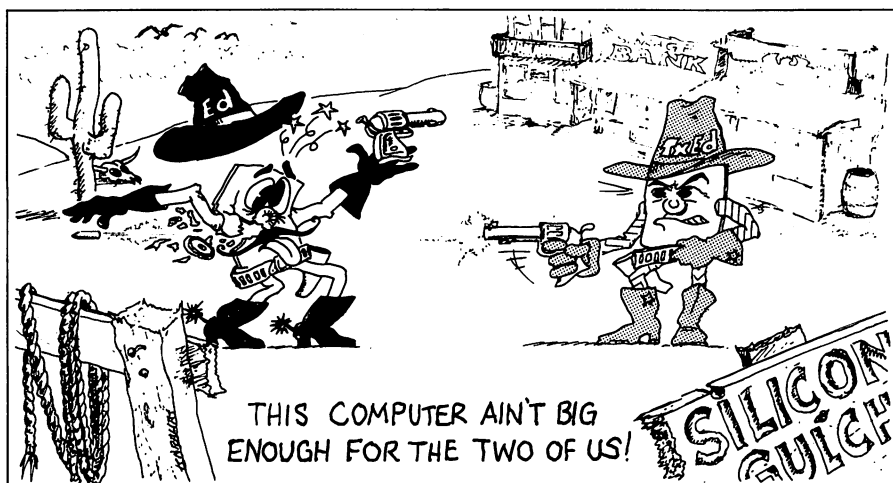
Additionally, the shards are all shaped the same, so there is no easy way to determine how they should fit into the puzzle. The inkblot keeps changing and it is possible to determine where the pieces should go by watching how the colors of your pieces change. This isn't easy when you only have a few pieces, so there is some help available. By performing a simple task and giving up some of your development bonus, Dr. Sigmund will place the piece for you. After all seven are placed you are transported back to the mind to repeat the entire process.

Although this explanation of the game-play may seem confusing, it is actually over-simplified. The game must be played to be understood and believed!

Product Specifics

Mind Walker has just about everything one would want in an Amiga game. The graphics are some of the best yet on the Amiga, and have not been done at the expense of action. Movement of all pieces is smooth, and shadowing is used effectively to produce a stunning three dimensional effect. The music is the best and most appropriate yet produced for a computer or video game. It is an eerie soundtrack which takes full advantage of Amiga sound through the use of panning stereo.

Commodore-Amiga and Synapse Software apparently weren't satisfied with just producing a quality piece of software. They decided to create a quality product, with a good manual and professional package design. The manual is clear and concise, with lots of colorful graphics. It walks you through the basics of the game and also hints at some of the deadly things in store for you that haven't been mentioned in this article.

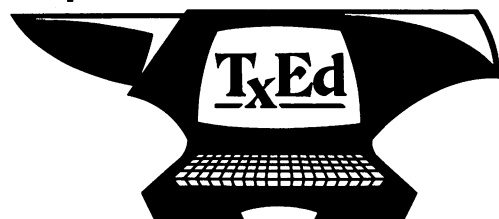


FIRE YOUR EDITOR.

And put Microsmiths' TxEd to use for you.

- ★ FAST display updates - more than TWENTY TIMES as fast as "Ed".
- ★ Designed from the ground up to take advantage of the Amiga user interface.
- ★ Multiple windows.
- ★ Very easy to learn, use menus for online help.
- ★ Simple, elegant set of commands.
- ★ Alternate command keys shown in menus allow fast command entry for experienced users.
- ★ Compact code works well with Amiga's multi-tasking.
- ★ The first Amiga directory requester that doesn't make you wait.
- ★ All around utility editor is good for programmers, and also useful as a simple word processor. Great for use with terminal programs.
- ★ Pronounced "Tex Ed" as in "Tex Ed, the Faster Editor in Silicon Gulch."

To order, send \$39.95 in check or money order plus \$2.50 postage and handling to: Microsmiths' TxEd, P.O. Box 561, Cambridge, MA 02140. Tel.: (617) 354-1224. Mass. residents add 5% sales tax. Amiga is a trademark of Commodore-Amiga, Inc. Designed by C. Heath. Dealer inquiries invited.



MICROSMITHS, INC.
P.O. Box 561, Cambridge, MA 02140



Lest you think that Mind Walker is the best thing since sliced bread, I do think that the program has a few shortcomings. First and foremost, the game is multitasking, but it takes most of the memory. It does operate from Workbench, so it is possible that limited multitasking may be accomplished with a bit of ingenuity, but only expansion memory makes this option realistic.

The second problem that the program has is that it is too easy to quit the game. All one has to do is hit the close window box and it's all over. It is easy to do this unintentionally because the box is located right next to the pause button.

The final problem that I have with the game is the intermission screens. They are poorly done, looking like something generated on an IBM. This lack of quality in an otherwise superior product cannot be excused.

Summary

Gaming enthusiasts should consider Mind Walker a must buy. Its combination of sound and graphics should set the standard by which other games are judged.

Mind Walker requires a single disk drive and 256 K of memory. It is not copy protected. Hopefully Commodore-Amiga is setting a good trend here by not copy-protecting a game. It has been my experience that most software pirates are willing to purchase the really good products. Mind Walker is such a product and should be purchased by everyone. It is especially important that Amiga owners strongly support the creation of quality products through their purchases. This will not only insure the production of other such works, but also help solidify the future of the Amiga itself.

•AC•

The *Alegra* Memory Board

"...the easiest, least obtrusive, and lowest-priced memory expansion available for the Amiga today."

**Reviewed by
Rick Wirth**

The Alegra memory expansion board by Access Associates provides the easiest, least obtrusive, and lowest-priced memory expansion available for the Amiga today.

Although the Alegra board has a very low retail price among its competition, do not think that it is a cheap or badly-designed board. Access Associates has provided a well-designed memory expansion board for the Amiga that has quality equal to that of major manufacturers of expansion boards for the IBM PC.

The board has four layers, a clean design, and bypass capacitors for noise reduction. Finally, no wires or jumpers are present, so no mistakes or last minute changes were made in the board layout. It is a board that is as fast as is possible, another sign of good design.

Wait states

Not all memory boards for the Amiga are of the "no wait state" design like this, some cause the processor to "wait" because the memory was not designed to be accessed at the maximum speed of the CPU.

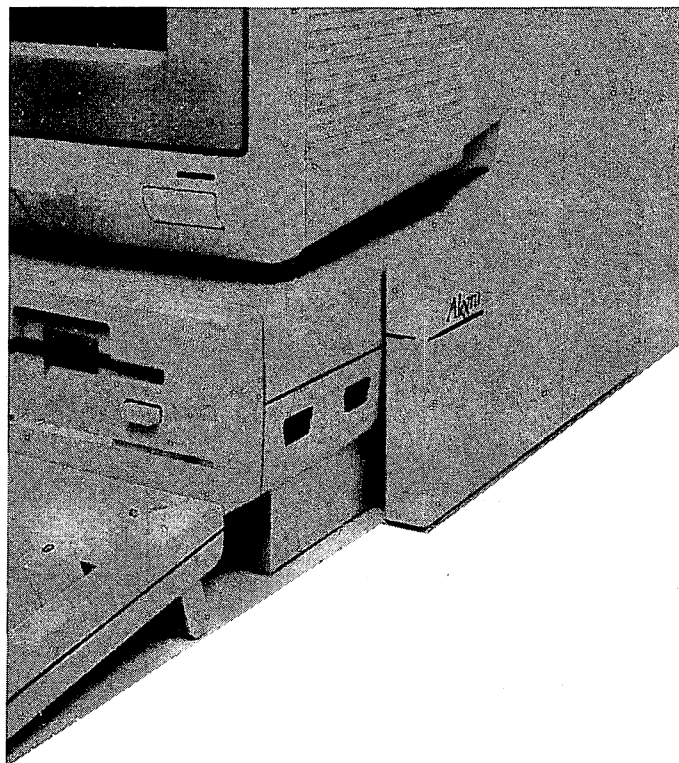
To verify that the Alegra board is of a zero wait state design, I ran a public domain memory speed test. This program was far from perfect, and provided numbers with little intrinsic value. The results can be used in a relative manner, as a ratio, not as an absolute benchmark between boards.

The internal, normal Amiga memory performed in 29,400,330 microseconds, while the Alegra memory took 29,116,968 microseconds, for an increase of 283,362 microseconds, in favor of the Alegra board.

Thus, the Alegra memory was faster than the memory in the Amiga. This can be expected because sometimes the processor in the Amiga must wait for the custom chips when it uses the memory built into the Amiga. However, the custom chips cannot use the external memory so the processor should never have to wait to use it.

This same program was also used to test the speed of another memory expansion board, which I will leave unnamed. This memory board took 30,650,306 microseconds, so the difference between them was 1,533,338 microseconds, in favor of the Alegra board.

The results show, in a relative sense, that this expansion board is slower than the Amiga, so one would assume this board has wait states.



Once again, these figures are relative, and cannot be used to compare speed. The Alegra is faster than both the normal Amiga memory and the other board, while the other board is slower than the memory already in the Amiga. That is all that can be extrapolated from these figures - nothing more.

The Alegra board can be purchased in two forms, the Alegra and the Alegra E. The standard Alegra board can only carry 512K bytes of memory, using 256K bit chips. The Alegra E comes with 512K byte of RAM, with additional sockets for the new one megabit memory chips. When these new chips are installed, the Alegra E becomes a two megabyte memory expansion board.

Unfortunately, one megabit memory chips are still far from cost effective, when compared to current 256K chip prices. Chances are, it will be at least a year before they are equivalent in price to the same amount of memory in 256K bit memories. Of course, when the price comes down, it will cost at least \$200 to upgrade to more memory.

Auto-configure

Another important design feature of the Alegra board is its automatic configuration ability. With the pre-release version of AmigaDOS 1.2, the memory in the Alegra board is added to the system memory without human intervention.

Why is this important? Non-auto-configuring boards require a modification of the startup-sequence on each and every Workbench disk you own. This modification invokes a special program to add the expansion memory to the system memory.

Auto-config is important because you do not have to install commands in the startup-sequence on every bootable disk you possess. Another reason is compatibility. If a board for the Amiga autoconfigures, then the Amiga can position it in the system memory map so that it does not conflict with other expansion boards. Without auto-configuration, you cannot be certain that any two devices will work together, even if they work independantly.

With AmigaDOS 1.2, installation is a breeze, simply remove the expansion cover, insert the Alegra board on the side, and turn on your Amiga. Since AmigaDOS 1.2 allows you to use the RAM: disk from the WorkBench as an icon, and also allows you to add the RAM: disk to the command path for the CLI, the Alegra board works especially well with AmigaDOS 1.2.

If you are still using AmigaDOS 1.1, you must add two commands to the startup-sequence on your WorkBench disk.

With a memory board, you can copy everything that AmigaDos needs from the Workbench disk into the RAM: disk, so you never have to see the "Please insert Workbench in any drive" requester again. Most of the programs you would use from your Workbench disk would only occupy about 40K of RAM disk memory. Unfortunately, many programs that run in a 512K machine need that 40K and I cannot recommend doing this without a memory expansion.

Small size

Externally, the Alegra board is very inconspicuous. It is only three fourths of an inch wide, the height of the Amiga, and roughly the same length as the external 3.5 inch disk drive. When attached, most people would even realize it is there. The case is a neutral beige that complements but does not perfectly match the Amiga's color, the external surface is a pebbled, flat finish that hides fingerprints.

My only objection with the Alegra board is its inability to pass the Amiga expansion port. If you buy the Alegra board, and plan to buy another expansion port device, that add-on must be a expansion bus pass-through, so that the Alegra board can remain the outermost add-on.

Obviously, if every manufacturer believes that someone else will be a bus pass-through, we will not be able to connect everything in the future.

Another objection is the lack of a battery backed clock, in either standard or optional form. It would have been a minimal increase in cost, but a greater feature, and it could have been done without increasing the size of the current board.

Therefore, if you are in the market for a low-cost, easy-to-install, and quality memory expansion to expand your Amiga, the Alegra board is for you. On the other hand, if you are looking for a memory expansion board with other capabilities, extensibility, and cost is no object, the Alegra board is probably not for you.

•AC•

Financial Manager

*Easy, powerful and fast single-entry accounting system for your Amiga. * Appropriate for home or small business. * Track your finances in up to forty accounts (checking, Visa, savings etc.) and 130 income/expense categories (groceries, clothing, car, paycheck, rental income etc.). * Reconcile your bank statement. * Establish a budget for all categories and graphically compare actual to budget amounts refining the budget as needed. * Print checks (any format). * Determine and print a networth statement. * Calculate/compare various loans and savings plans for best terms. * Customize the integrated tax-calculation/projection to suit your own needs and the new tax laws. * Track special occasions (with optional reminder of coming events whenever you boot up). * Manage a list of names, addresses and phone numbers. * Summon and select context-sensitive help/options throughout the program using the keyboard or mouse. * Program constantly anticipates user's intentions and provides a guessed answer which can be accepted with a single touch (keyboard or mouse) or typed over, drastically reducing typing. * Requires 512K bytes memory and one disk drive.*

We're so sure you'll love it that it comes with a **30-Day Money-Back Guarantee!**

(When purchased directly from Marksman Technology Inc.)

Only \$89.⁹⁵ Including postage and handling.

Available immediately - we've been shipping since July!

VISA

MasterCard

800-334-7792 (Orders)

Marksman Technology Inc.
Route 5, Box 221A
Santa Fe, NM 87501
505-455-2681 (Inquiries)

PHASAR™

Marksman
TECHNOLOGY INC.

TxEd

Version 1.3

"...a very good editor and an excellent value."

**Reviewed by
Jan and Cliff Kent**

TxEd is a programmer's text editor, intended primarily for the preparation of the ASCII text files used as input for a language compiler or an assembler. It is a very good editor and an excellent value. Although it is not sold as a word processor, TxEd can also serve well in that capacity for many users. This review describes version 1.3 of TxEd which includes several significant improvements over earlier versions.

We received our copy of TxEd by mail 5 business days after mailing the order. The \$59.95 package includes a disk which is not copy-protected, and a 19 page manual that measures 5 1/2 by 7 1/2 inches. It is readable, complete, and clear but the small type is distracting when trying to find a specific reference to a command. However, since the pull-down menus are well-designed, we did not find ourselves referring to the manual very often. For a good laugh, be sure to read the Copyright Notice on page 1.

You can start TxEd from the Workbench with a double click of its icon, or from the CLI using a conventional command line with options.

If you start the program from the CLI, two command line options are available. The -f option opens a borderless window that displays 23 lines of 80 characters per line, useful when editing program source from other environments. And you may add a file name to the command line to automatically load a file for editing. Two of the functions on the RANDOM menu are available only if TxEd is started from the CLI. NEW CLI starts up a fresh AmigaDOS CLI window allowing access to any AmigaDOS command without exiting TxEd. This is normally the fastest way to run a compiler or assembler during program development. MORE TxEd starts up a new copy of TxEd. You can use the new copy to edit include files or to find an old piece of code that you need in a new program.

The text file to be edited is loaded into RAM for editing. Many editing functions are unusually fast as a result. The minimum text buffer size is 35k, but is increased automatically when loading larger files from disk.

All of TxEd's commands are available from the 5 pull down menus. Most commands are duplicated on the keyboard by a combination of the right Amiga key and a printable character key. The keyboard commands are generally well-chosen and, with the menus available as HELP, we were both able to learn TxEd in a very short time.

The PROJECT menu contains commands that will clear the text buffer, open/create text files, save the text buffer contents to disk files, and exit TxEd. No surprises here. The prompts are clear and disk accesses fast. If you try to exit TxEd without saving the edited version, a requester will question your intent. A welcome change from other Amiga programs is that you don't have to wait until the disk directory search has completed before entering a drawer or file name. If you know what you want, just click the DRAWER or FILE string gadget and start typing. TxEd will accept keyboard input during the directory search. As a drawer fills with files this can become a big timesaver.

The EDIT menu commands allow you to mark a block, then either copy or cut to the Amiga clipboard device. Text from the clipboard device may also be inserted into the text buffer. Since the clipboard device is a system resource it is possible to import/export text from/to other programs running as independent tasks, including other copies of TxEd. Many complex editing functions that are possible, but potentially confusing, in earlier multi-window editors can be accomplished almost by instinct by starting several copies of TxEd, loading the needed files, and moving blocks between them via the Amiga Clipboard. New in version 1.3 is the ability to print the contents of the clipboard in the background while you continue to edit in the TxEd window. As with most AmigaDOS functions the clipboard contents can also be "printed" to a disk file.

The EDIT menu also has commands to delete a line, delete from the cursor to the end of the line, and recall a deleted line. Some days this last function is very nice. The BACK SPACE key deletes as it moves the cursor left. The DEL key deletes the character under the cursor.

The edit cursor can be moved with the keyboard arrow keys, or by pointing with the mouse and clicking, or by holding down the left mouse button and dragging the cursor. If you drag the cursor to the top or bottom of the screen, the entire display will scroll. In addition, there are commands for moving forward and backward a word at a time, moving to the left or right end of the current line, moving to the top or bottom of the document, scrolling the display by half pages (14 lines), and jumping directly to any line by number.

As expected, the RETURN moves the cursor to the left margin and starts a new line. The ENTER key (on the numeric keypad) is slightly different. In addition to starting a new line, it auto-indents the cursor for structured code entry, or if the line above was a

GRIDIRON!™
The Football Simulator

Eight Second play, fourth and goal on the one yard line and you're down by five. No room for mistakes. Time for that special play you created. You select it and snap the ball. The quarterback hands off to the halfback who sweeps right, spins and fires a bullet across the field back to the quarterback in the end zone. It's good! The buzzer sounds and the crowd goes wild. You won!!

THIS IS GRIDIRON!™ the most exciting football game ever made! GRIDIRON!™ is first to combine arcade graphics and stereo sound with computer simulation. Create your own winning plays, or pick from over 400 in the standard play book. **WARNING: THIS GAME IS HABIT FORMING!**

Created by football addicts for football addicts!

GRIDIRON!™: \$29.95 (1-2 players)

A quality product from **Bethesda Softworks**.

Visa and MasterCard orders call toll free: (800) 992-4009
 Check or money orders: P.O. Box 1153, Bethesda, MD 20817
 Add \$2.50 for shipping, DC, MD, VA residents add 5% tax

—Dealer Inquiries Invited—
 Bethesda Softworks, 9208 Burning Tree Rd, Bethesda, MD 20817

legal comment for C or the Amiga Assembler, it copies the appropriate comment introduction characters to the new line.

The most striking thing about the cursor movement commands is the display speed. This is the class of performance that we were looking for from the Amiga. Using half page scrolling you must be careful not to scroll right past the spot you were looking for.

The SEARCH menu has SEARCH, REPLACE, and REPEAT last search or replace. Switch gadgets allow you to search forward or backward, recognize or ignore case, and do global replace with or without individual prompts. Like TxEd's display functions, these are very fast. (SEARCH and REPLACE locate the designated string quickly enough to be disconcerting to an Amiga novice accustomed to Word Star. BEGINNING/END of File seems to take about as long as BEGINNING/END of Line, even when dealing with a 135k file. - Jan)

The RANDOM menu has some nice extras, including control of display colors, numeric entry of non-keyboard characters, a strip command (that removes carriage returns from non-Amiga text files), word wrap on/off, paragraph reformat, the insert/overstrike toggle, as well as the new CLI and more TxEd functions mentioned earlier.

Word wrap and paragraph reformat brings up the subject of using TxEd as an editor for word processing. When word wrap is on, TxEd will move the correct characters down to the next line as expected. The line width used is the physical width of the window as set with the window sizing gadget. Paragraph reformat is

equally simple. Text is reformatted as part of the same paragraph until a blank line or a line with leading blanks or tabs is found. Again, the current window width determines the line length. Not fancy, but very fast.

Although TxEd has no sophisticated print formatting commands, the distribution disk includes the public domain program PROFF, a text formatter by Ozan Yetti of York University. PROFF is similar to FORMAT, ROFF, and SCRIPT. It reads a text file with format commands embedded in it and writes the formatted text to video, the printer, or to a second disk file. This combination of programs is not a simple, instinctive, what you see is what you get, word processor. But PROFF does have a very wide range of formatting commands, including page headers and footers, semi-automatic table of contents, and a macro language that can be used to handle special cases. Also, having separate editor and print formatting programs running as separate tasks gives the very best kind of print spooling.

TxEd alone works well for simple letter writing. With TxEd and PROFF we would not hesitate to prepare a program's manual, or a software review. This review was written and edited with TxEd, then formatted and printed with PROFF. It worked very well.

We do have some small complaints. The commands that move the cursor a word at a time place the cursor on the space between words rather than on the first letter of each word. This is seldom the right place, so another keystroke is required. Also, punctuation is ignored in the move by word. As a result, move by word can't be used to move to the beginning or end of a C language comment or to the start of the first word on the line.

TxEd makes good use of the Amiga system environment to be an effective text editor without requiring a large, hard-to-master command set. But two commands are missing. TxEd has no delete word command. (I have never been without this command before and I miss it. - Cliff) Also, there is no way to scroll the display a few lines without moving the cursor. The ability to move the cursor and scroll the screen easily with the mouse nearly replaces a simple screen scroll, but it's usually faster to keep your hands on the keyboard.

TxEd is a fairly simple editor that is so fast that it seldom leaves you waiting for something to finish. (I'm convinced that all clever, complicated new editing commands are devised by programmers while waiting for their old editor to finish doing something or another. - Cliff) TxEd is being actively improved and upgraded by Microsmiths, so there should be no fear of being stuck with an early version. Stop wasting time trying to learn ED's dozens of commands. If you program in C, Pascal, or Assembler (or just want to edit your Startup-Sequence file or write some letters), get TxEd.

About The Authors

Jan and Cliff Kent run Kent Engineering & Design, providing custom digital hardware and software design. Cliff has written an 83 Standard Forth compiler for the Amiga for in-house use to facilitate software development on the Amiga. The first, a telecommunications package called MacroModem, is available now. His article on F83 Forth String Functions appeared in the March/April 1986 issue of Forth Dimensions.

•AC•

Amazing Directory of Amiga Products

Software

Entertainment

Access Software			
Leaderboard Golf	\$39.95	Archon II: Adept	\$49.95
Accolade		Artic Fox	\$39.95
Mean 18	\$44.95	Autoduel	
Activision/ Infocom		Chessmaster 2000	\$44.95
A Mind Forever Voyaging	\$44.95	Deluxe Music Construction Set	
Ballyhoo	\$49.95	Deluxe Paint	\$79.95
Borrowed Time	\$44.95	Golden Oldies	\$34.95
Brataccas	\$49.95	Marble Madness	\$39.95
Championship Baseball	\$44.95	One on One	\$39.95
Championship Golf	\$44.95	Seven Cities of Gold	\$39.95
Cutthroats	\$39.95	Skyfox	\$39.95
Deadline	\$49.95	Software Golden Oldies	\$34.95
Enchanter	\$39.95	The Bard's Tale	
GameMaker	\$59.95	Ultima III	\$59.95
GameMaker Libraries	\$34.95	Epyx Inc.	
GBA Champ. Basketball	\$44.95	Temple of Apshai Trilogy	\$39.95
Hacker	\$44.95	Geodesic Publications	
Hacker 2	\$49.95	Triclops Invasion	\$21.00
Hitchhiker's Guide to the Galaxy	\$39.95	Jenday Software	
Infidel	\$44.95	Conversation with a Computer	\$29.50
Little Computer People Discovery Kit	\$49.95	Megatronics	
Mindshadow	\$44.95	Talking Trivia	\$19.95
Planetfall	\$39.95	Meridian Software	
Portal	\$44.95	Games Gallery	\$29.95
Ractor	\$44.95	MicroIllusions	
Seastalker	\$39.95	The Faery Tale	\$39.95
Shanghai	\$44.95	Mindscape	
Sorcerer	\$44.99	Balance of Power	\$49.95
Spellbreaker	\$49.95	Brataccus	\$49.95
Starcross	\$49.95	Deja Vu	\$49.95
Suspect	\$44.95	Haley's Project	\$44.95
Suspended	\$49.95	Keyboard Cadet	\$39.95
Tass Times in Tonetown	\$44.95	Master Type	\$39.95
The Music Studio	\$59.95	Racter	\$44.95
The Witness	\$39.95	The Halley Project	\$44.95
Wishbringer	\$39.95	The Perfect Score	\$79.95
Zork I	\$39.95	Defender of the Crown	
Zork II	\$44.95	S-D-I	
Zork III	\$44.95	Sinbad and the Throne of the Falcon	
The Music Studio		The King of Chicago	
Baudville		Polarware	
Video Vegas	\$34.95	Oo-Topos	\$29.95
Bethesda Softworks		The Coveted Mirror	\$29.95
Gridiron	\$69.95	The Crimson Crown	\$29.95
Commodore Business Machines		Transylvania	\$29.95
Mindwalker	\$49.95	Sierra On-Line Inc.	
Computer Solutions		King's Quest I, II, III	\$49.95
Strategy and Adventure	\$11.95	The Black Cauldron	\$39.95
Dark Horse		Strategic Simulations	
Chessmate	\$32.50	Computer Baseball	\$39.95
Electronic Arts		SubLOGIC	
Adventure Construction Set	\$39.95	Flight Simulator II	\$49.95
Archon	\$39.95	Jet	\$49.95

NEW! PUT YOURSELF IN THE 21st CENTURY

Experiment with **Artificial Intelligence!** We supply the **Expert System Kit** including a knowledge base editor, an automatic trainer, a run-time driver, and examples. You supply the creative imagination.

Complete instruction manual guides you to creating your own application. The knowledge base editor lets your application exchange data with other programs and even run DOS commands and other programs. Included is a sample application that talks! The automatic trainer analyzes your data and learns to draw the correct conclusions. The driver lets you run your expert system standalone. Think of the applications! We will be supporting this kit with a newsletter so you can share knowledge bases, techniques and ideas.

Don't wait any longer! You can get in on this exciting new field for only **\$69.95**.

HOW TO WRITE BETTER PROGRAMS? EASY!

The key is having the right tools. The **EXPLORER** disassembling debug monitor steps you through code and shows the 68000's registers as they change. You have complete control as you single step or set breakpoints. Have a live window onto memory. Watch what other tasks are doing. See where you are with an on-line memory map. Learn from code others have written by disassembling it, capturing it, and changing it to suit you!

The **EXPLORER** has other powerful features: display memory and files in Hex and ASCII, memory modify, search, move, fill, display and change registers, disassembly trace, load programs, disassemble to disk. Output to printer or disk file. Powerful commands: command loops, time-saving macros, logging to disk, text display, real-time RAM view, & more!

The **EXPLORER** is just what you need to take control of your program development, and at **\$49.95** it is a solid value too!

If your **AMIGA** dealer is out of stock on the **EXPLORER** or the **Expert System Kit** you may order from us directly. Shipping & handling \$4. COD add \$4. Orders: **(800) 328-8322 Ext. 804**. Info: (612) 559-6601. Money orders or checks to:

INTERACTIVE ANALYTIC NODE

Interactive Analytic Node
2345 West Medicine Lake Drive
Minneapolis, Minnesota 55441

Dealer and distributor enquiries welcome.
AMIGA is a trademark of Commodore-Amiga Inc.

The Other Valley Software	
Monkey Business	\$24.95
The Quality Cottage	
The Orator	\$39.95
Firebird Software	
The Pawn	\$44.95

Business

Aegis Development	
Impact	\$199.95
Allen Computer Technologies	
Hyperbase	\$79.97
Batteries Included	
BTS The Spreadsheet	\$69.95
Isgur Portfolio	\$199.95
PaperClip Elite	\$129.95
Thunder	\$199.95
B.E.S.T. Inc.	
B.E.S.T. Financial	\$395.00
Byte by Byte	
Financial Plus	\$295.00
Infominder	\$89.95
Write Hand	\$50.00
Chang Labs	
General Business 3-Pak	\$499.50
Rags to Riches/ Accounts Payable	\$199.95
Rags to Riches/ Accounts Receivable	\$199.95
Rags to Riches/ General Ledger	\$199.95
Rags to Riches/ Payroll	\$199.95
Rags to Riches/ Sales Analysis	\$199.95
Clockwork Computers	
CCI Bottom Liner A	
CCI Integrated Merchandisers Sys.	\$650.00
Colony Software	
CashCraft!	\$99.95
FileCraft!	\$79.95
MailCraft!	\$25.00
Commodore	
Textcraft	\$99.95
Textcraft Plus	
Computer Solutions	
Finance I	\$11.95
Computerware	
Accounts Payable	\$99.00
Accounts Receivable	\$99.00
Check Ledger	\$99.00
General Ledger	\$99.00
Inventory Control	\$99.00
Payroll	\$99.00
Developers of Advanced Software	
D.A.S. Business Finance	\$30.00
D.A.S. Home Finance	\$30.00
Digital Creations	
Gizmoz	\$49.95
Digital Solutions	
LPD Filer	\$150.00
LPD Planner	\$150.00
LPD Writer	\$150.00
Eastern Telecom, Inc.	
INFO+	\$49.99
Eclipse Data Management	
Rent-Pro	\$2,000.00
For-Trac	\$2,250.00
Electronic Arts	
Financial Cookbook	\$49.95
Ensign Software	
Commodity Futures Real-time Tic Chart	\$795.00
Foxware	
FoxFile	\$50.00
Gander Software	
Financial Planner	\$100.00
Program Generator	\$400.00

The Data System	\$200.00
Time & Task Planner	\$100.00
Getting Enterprises Inc	
File Clerk I	\$100.00
Funtime I	\$50.00
HobKit I	\$125.00
Grey Associates	
Disk Traffic Controller	
Harvsoft	
Info Base	\$45.00
HC Software Australia	
Amiga Record Manager	\$38.00
INSIGHT/Lehner Communications	
The Financial Time Machine	\$59.95
Lattice	
Unicalc	\$79.95
Lionheart Business Software	
Business Statistics	\$145.00
Decision Analysis Techniques	\$110.00
Experimental Statistics	\$145.00
Explanatory Data Analysis	\$75.00
Forecasting and Time-Series	\$145.00
Linear and Non-linear Programming	\$95.00
Multi-Variate Analysis	\$145.00
Multivariate Analysis	\$145.00
Optimization	\$110.00
Pert & Critical Path Techniques	\$95.00
Quality Control and Industrial Experiments	\$145.00
Sales and Market Forecasting	\$145.00
Marksman Tech Inc.	
PHASAR	\$89.95
Maxisoft/Electronic Arts	
MaxiBase	\$150.00
MaxiComm	\$49.95
MaxiDesk	\$70.00
MaxiPlan	
Maxipower Series	\$29.95
Megasoft Limited	
A Filer/ A Report	\$59.95
Megatronics	
Merge Master	\$24.95
Spellcraft	\$24.95
Micro-Systems Software	
Organize!	\$99.95
Scribble!	\$99.95
Analyze!	\$99.00
New Horizons Software	
Flow	\$99.95
Northeast Software Group	
ORDER	\$44.95
Olamic Systems Corporation	
2 + 2 Home Management System	\$99.00
PAR Software	
Real 1	\$149.00
Biz I	\$149.00
Home I	\$69.00
Progressive Peripherals and Software	
Logistix	\$249.95
RTL Programming Aids	
Wordwright	\$75.00
Scientific Software	
Equation Plotter	\$29.95
Sedonna Software	
Money Mentor	\$99.95
Softwood Co.	
MiAmiga File	\$99.95
MiAmiga Inventory	
MiAmiga Investor	
MiAmiga Ledger	\$99.95
MiAmiga Payables	
MiAmiga Recievables	
MiAmiga Word	\$99.95

The Efficient Way To Communicate

MacroModem 1.2TM Pioneers User-Written Menus

User-defined macro command sets and companion help screens with "point & click" or keyboard operation

- One macro can return hundreds of key codes.
- Automate a session with normal command keystrokes stored in macros.
- User-written macro commands can invoke virtually anything MacroModem can do.

Operate a remote system almost entirely with the mouse

by writing macro command sets that mimic the menus and commands of a remote system

Special features include:

- 20 commands on function keys
- Multi-windows, multi-tasking
- Fast terminal window display
- 10 line Compose Message window
- Separate FileFilter program

\$69.95

Dealers may order MacroModem from Southern Technologies
800/647-7741

Kent Engineering & Design
P.O. Box 178, Mottville, N.Y. 13119
(315) 685-8237



The bridge to your computing future.

MacroModem & MacroWare are trademarks of Kent Engineering & Design.
Amiga is a trademark of Commodore-Amiga, Inc.

Nancy™

A Spelling Checker

FOR THE COMMODORE AMIGA

RETAIL PRICE: \$60.00

FEATURES:

- recognizes over 80,000 words
- suggests corrections
- uses speech
- automatic correction of common errors
- learns from your mistakes
- user defined dictionaries
- online help function
- adds new words automatically
- works with most popular file editors
- like Amiga Textcraft™ and Scribble!™

TERMS: PREPAID (CHECK OR MONEY ORDER)

THE COMPUTER CLUB
4843A SOUTH 28TH STREET
ARLINGTON, VIRGINIA 22206
(703) 998-7588

© Copyright 1986 by the Computer Club, All Rights Reserved
Nancy is a Trademark of The Computer Club
Amiga Textcraft is a Trademark of Commodore-Amiga, Inc.
Scribble! is a Trademark of Micro Systems Software

(DEALER INQUIRIES INVITED)

A-TALK™

Advanced Communication and Terminal Program for the AMIGA

- **KERMIT - XMODEM - XMODEM/CRC - ASCII -**
Xmodem binary files are stripped of padding characters.
- **DIAL-A-TALK** - Phone directory, redial and script language for auto-login. Programmable function keys.
- **VT100/VT52/H19/ANSI/TTY EMULATION** - Full emulations including 132-char/line mode and graphics fonts.
- **MULTI-TASKING SPOOLER** - For concurrent printing and saving of files during your terminal session.
- **VOICE OPTION** - For having mail read aloud and for telling you how the call and login are progressing.
- **SETTINGS** - Over 10 modem types supported. All communication parameters, including X-on/X-off.

A-TALK lists for \$49.95 and is not copy protected.
\$2.00 shipping; CA residents add 6.5% sales tax.

Felsina Software
3175 South Hoover Street, #275
Los Angeles, CA 90007
(213) 747-8498

Softworks Basic	\$99.00
Software Group/Amiga	
Enable	
T.R. Software	
E.T. Writer	\$49.95
The Computer Club	
Nancy	\$60.00
The Other Guys	
AMT	\$39.99
KEEP-Trak Accounts Payable	\$79.99
KEEP-Trak Accounts Receivable	\$79.99
KEEP-Trak General Ledger	\$129.99
KEEP-Trak Inventory	\$79.99
KEEP-Trak Payroll	\$79.99
Omega File	\$79.99
Spell-man, Add-man	\$39.99
VIP Technologies	
VIP Analysis	
VIP Consultant	
VIP Forethought	
VIP Professional	\$199.95
WordPerfect Corp.	
WordPerfect	
X-Scope Enterprises	
Write-To-Left	\$29.95
Educational	
Academy Software, Inc.	
Typing Tutor And Word Invaders	\$34.95
CBS Interactive Learning	
Mastering the SAT	\$99.95
Deskware	
Sci-Calc	\$19.95
Finally Software	
Dr. Xes	\$49.95
Senor Tutor	\$49.95
Talker	\$69.95
Trendwriter	\$99.95
First Byte	
Kidtalk	\$59.95
Speller Bee	\$59.95
JMH Software of Minnesota Inc.	
The Talking Coloring Book	\$29.95
Micro Prose	
Gunship	\$39.95
MicroEd Inc.	
Across the Plains	\$59.95
Basic Grammer	\$29.95
Beginning Reading Series	\$119.80
Capitalization	\$29.95
Elem. Social Studies	\$29.95
Great Lakes Fur Trade	\$79.95
High-level Vocabulary	\$49.95
Lewis & Clark Expedition	\$89.95
Making Our Constitution	\$79.95
Point	\$29.95
Punctuation	\$29.95
Spelling Detective Game	\$39.95
Spelling Series	\$149.80
Word Demons	\$29.95
Basic Grammar	\$29.95
Beginning Reading Two	\$29.95
Capitalization	\$29.95
Punctuation	\$29.95
Social Studies Vocabulary (El.)	\$29.95
Spelling Level Five	\$29.95
Spelling Level Four	\$29.95
Spelling Level Six	\$29.95
Spelling Level Three	\$29.95
Spelling Level Two	\$29.95
Vocabulary Series (High Level)	\$49.95
Word Demons (usage)	\$29.95

"Real Soon Now..."

...words that you will not hear from ASDG

Available NOW!

ASDG experienced some of the same problems other manufacturers have had with production delays. What sets ASDG apart is how we have dealt with our problems. We shipped temporary loaners. In some cases, we have volunteered money back for being behind. In every case we kept our customers completely and accurately informed. ASDG's responsibility to you doesn't end when we cash your check. Rather...that's where it begins.

Mini-Rack-D

a 2 slot totally ZORRO compatible backplane. Powered at +5, -5 and +12 volts. All metal closure. Measures 6"w x 10"h x 10"d. **List \$325 with ASDG memory board.** Covered by *Mini-Rack-Buy-Back*.

Mini-Rack-C

a 2 slot ZORRO subset. Powered at +5 volts. All metal enclosure. Measures 6"w x 10"h x 10"d. **List \$195 with ASDG memory board.** Covered by *Mini-Rack-Buy-Back*.

Recoverable Ram Disk

Software alternative to M:. Files copied to ASDG recoverable ram disk are preserved through resets and crashes. Totally AmigaDOS compatible. Dynamically manages ram disk memory. **FREE with ASDG memory board.**

.5M, 1M and 2M FAST RAM Boards

All ZORRO compatible FAST ram boards. Autoconfiguring. No wait states. .5M (1/2 mbyte) and 1M (1 mbyte) user expandable to 2M (2 mbytes). Optional socketing (\$75) makes upgrade completely solderless. 1 year warranty.

	List	Introductory Price	Introductory User Group Price
.5M 1/2 Mbyte of FAST Ram:	\$450.00	\$395.00	\$370.00
1M 1Mbyte of FAST Ram:	\$650.00	\$575.00	\$550.00
2M 2Mbytes of FAST Ram:	\$900.00	\$795.00	\$750.00

ASDG Incorporated

280 River Rd. Suite #54A
Piscataway N.J. 08854
(201) 540-9670

New Jersey residents please add six percent sales tax. ASDG will pay UPS standard delivery in the continental U.S.
Dealer and VMR pricing available. Demonstrations to large user groups.
SYSOPS—Call us for more information about special pricing.

AMIGA CUSTOM PRINTER DRIVER:\$35+S/H

Create your own printer driver for
virtually any printer.

• MENU DRIVEN • WORKS THRU PREFERENCES

We are world famous for our selection
of Amiga software!

Call our Amiga BBS at night or call
during store hours to order!

We specialize in
COMMODORE AND AMIGA COMPUTERS
SOFTWARE SUPERMARKET
31621/2 Delaware Ave.
Kenmore, N.Y. 14217

For Dealers only:
Please call for pricing
on our printer driver.



(716)873-5321
& THE PRINTER STOREhouse

Rescue Your Recipes!



Compu Cuisine
Over 200 Recipes
in 8 categories.

Edit and add own recipes.

Organize and search files
by category or ingredient.

Send check or money order for

for the Amiga! \$29.95 to:
Adept Software
P.O. Box 700702
San Jose, Ca. 95170

MicroIllusions

Discovery Math \$39.95
Discovery Spelling \$39.95

Queue Inc.

American History Adventure \$59.95
Analogies I \$65.00
Analogies II \$65.00
Antonyms \$34.95
College Aptitude Reading Comprehension \$65.00
Comprehensive Grammar Review I \$54.95
Comprehensive Grammar Review II \$54.95
How a Bill Becomes A Law \$59.95
Lessons in Reading & Reasoning I \$39.95
Lessons in Reading & Reasoning II \$39.95
Lessons in Reading & Reasoning III \$39.95
Lessons in Reading & Reasoning IV \$59.95
Lessons in Reading & Reasoning Package \$149.95
Practice Composition I \$44.95
Practice Composition II \$54.95
Practice Composition III \$44.95
Practice Composition IV \$44.95
Practice Composition Pkg I (inc. I, II, & III) \$144.95
Practice Composition Pkg I (inc. IV & V) \$85.00
Practice Composition V \$44.95
Reading & Thinking I -Grades 2-3 \$54.95
Reading & Thinking II -Grades 4-5 \$54.95
Reading & Thinking III -Grades 6-8 \$54.95
Reading Adventure I -Grades 2-3 \$39.95
Reading Adventure III -Grades 6-8 \$59.95
Sentence Completion \$44.95
Spanish Grammar I \$34.95
Spanish Grammar II \$34.95
Spanish Grammar III \$34.95
Starting A New Business \$59.95
U.S. Geography Adventure \$59.95
Vocabulary Adventure I -Grades 6-8 \$59.95
Vocabulary Adventure II -Grades 8-10 \$59.95
Vocabulary Adventure III -Grades 10-12 \$59.95
World Geography Adventure I - The Americas \$59.95
World Geography Adventure II - Europe \$59.95
World Geography Adventure III - Africa \$59.95
World Geography Adventure IV - Asia \$59.95
World History Adventure \$59.95
Analogies \$65.00
Antonyms \$34.95
Clear Sentences \$54.95
Comprehensive Grammar \$54.95
French Grammar \$34.95

Scarborough Systems Inc.

MasterType \$39.95
The Dragon Group
Amiga Coloring Books \$34.95
The Other Guys
Great States U.S.A. \$39.95
Unicorn Software Co.
Decimal Dungeon \$49.95
Fraction Action \$49.95
Math Wizard \$49.95

Graphics

Aegis Development
Aegis Animator \$139.95
Aegis Art Pak #1 \$34.95
Aegis Art Pak #2 \$34.95
Aegis Draw \$199.95
Aegis Images \$79.95
Aegis Impact \$199.95
Degas Elite \$79.95
Diga!
Associated Computer Services
Weather Graphics \$995.00

\$59.95

Commodore Business Machines	
Graphicraft	\$49.95
Computer Food Inc.	
3-D Graphics Library	\$100.00
DeluxeHelp Inc.	
DeluxeHelp for Deluxe Paint	\$19.95
Digitek Software Development	
Deluxe Clip Art #1	\$29.95
Crystal Rose Software	
Analytic Art	\$59.95
Discovery Software	
Grabbit	\$29.95
Electronic Arts	
DPrint Art (Volume 2)	\$29.95
Deluxe Video	\$99.95
DPaint Art/Utility Disk	\$29.95
New Tech. Coloring Book	\$19.95
Film Production Toolkit	
Film Production Toolkit	\$1,500.00
Grey Associates	
Art Director	
Interactive Microsystems	
MediaPhile	\$197.00
JDK Images	
PRO VIDEO CG I	\$1,500.00
Microillusions	
Dynamic-CAD	\$499.95
S. Anthony Studios	
Laser Up! Graphics	\$79.95
Laser Utilities Vol. 1	\$39.95
LaserFonts 1	\$39.95
Scott Lamb	
Sprite/Graphics Editor	\$50.00
Soft Circuits Inc.	
PCB CAD	\$1,024.00
PCB-CAD	\$1,000.00
PCLO	\$1,000.00
Zuma Group, Inc.	
TV*FONTS	\$29.95
TV*TXT	\$24.95
Languages and Utilities	
Absoft	
AC Fortran	\$295.00
AC/Basic	
Fortran 77	\$295.00
Capilano Computer Systems	
LogicWorks	\$199.95
Hex Utilities	\$49.95
Logic Compiler	\$129.95
Central Coast Software	
DOS-2-DOS	\$55.00
Classic Image	
Diablo	\$29.95
Classic Image Software	
Disk Library	\$49.95
Creative Solutions	
Multi-Forth	\$179.00
Commodore Business Machines	
Amiga C	\$149.95
Amiga LISP	\$199.95
Amiga Macro Assembler	\$99.95
Amiga Pascal	\$99.95
Amiga TLC Logo	\$99.95
WACK Software Toolkit	\$99.95
Amiga Transformer	\$99.95
The Mirror "Hacker" Package	\$49.95
The Mirror Disk Copier	\$49.95
Data Dynamics, Inc.	
Genie	
Data Research Processing Inc.	
The Key to "C"	\$34.95

Expand your mind
as you expand your Amiga's
memory...
with Promiga's
MegaBoard 2

Promiga's powerful and compact
MegaBoard 2 adds 2 full
megabytes of RAM to your Amiga.
You can develop more sophisticated programs,
build larger files, and run all Amiga software quicker.
And it's amazingly affordable.
Suggested retail price: \$695.

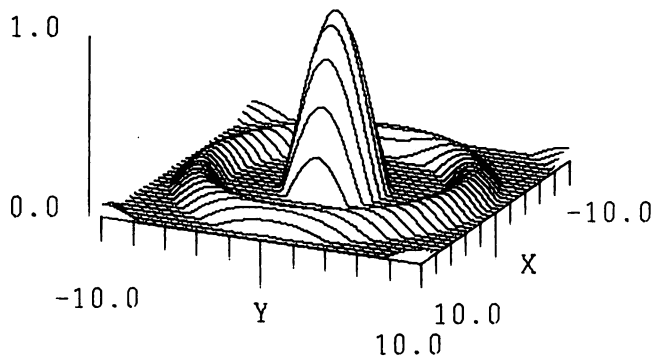
Now the only limitation to what
you can do with your Amiga is your mind,
not your Amiga's memory.

For order information, contact
Promiga
(602) 375-1206
P.O. Box 61834
Phoenix, AZ 85082-1834

NOW! SCIENTIFIC PLOTTING FOR THE AMIGA

- **Silver-Reed EB 50 Printer/Plotter** \$299.95
Parallel I/F for the Amiga
4 Color Pen Plotter and Printer
Built-in Business Graphics
Limited Supply Available
- **SCILOT—Software for the Amiga** \$89.95
2D and 3D Scientific Plotting
Other Device Interfaces Available
- **Package Deal—both for** \$359.95

Plot of $\text{Sinc}[\text{SQRT}(X^2 + Y^2)]$



Make check or money order to:
OLI P.O. Box 41122
Plymouth, MN 55447 (612) 888-5358
Add \$4 for shipping and handling.
Prices subject to change without notice.

CSA's

AWARD WINNING

32 BIT TURBO AMIGA

Acclaimed to be the most powerful, innovative expansion system for the **AMIGA** Personal Computer. **10 times Faster than a standard AMIGA.**

We are the only ones to offer mainframe speed and... power that will sit right on your desk and run all AMIGA software including FORTRAN 77, C-Compilers and Assemblers. Now, that's amazing computing power.

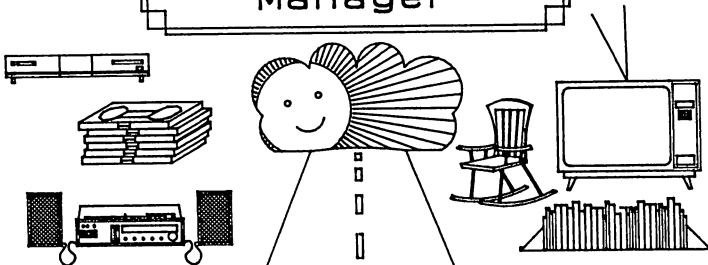
- 5-slot Expansion Cabinet
- SCSI Host Adapter
- 69020/68881 14 Mhz CPU
- 20MB Winchester Hard Disk
- 512K Static RAM
- ...Call or write today:

COMPUTER SYSTEM ASSOCIATES, INC.
7564 Trade Street
San Diego, CA 92121
619-566-3911

Amiga Home Inventory Manager

\$34.95

\$34.95



The Road to "Friendly" Software

Did you ever wonder how much money is tied up in inventory in your home? Did you ever search through insurance policies looking for that Renters/Homeowners policy inventory sheet? If so then this software is for you!

- soon to be released: antique cataloger, audio cataloger, video cataloger, stamp cataloger, coin cataloger.
- covers insurance information.
- plus many more features.
- sort or search any of the pre-defined fields.
- multiple report forms.
- 100% use of Amiga interface.
- covers serial #'s, purchase date and place.

100% purchase price refund if not COMPLETELY satisfied!
Add \$3.50 for shipping & handling

SUNSMILE SOFTWARE
533 FARGO AVE.
BUFFALO, NY 14213
716-885-5670

Delta Research	
J-Forth	\$200.00
Desktop A.I.	
dBx Translator	\$1,000.00
Digital Creations	
Digital Link	\$69.95
Image Writer II Driver	\$49.95
Discovery Software	
Marauder	\$39.95
Eclipse Data Management	
KWIK-SPEAK I	\$50.00
Electronic Arts	
Amiga Programmer's Library	\$200.00
Gimpel Software	
Amiga-Lint	\$98.00
Interactive Analytic Node	
Expert System Kit	\$69.95
The Explorer	\$49.95
Jenday Software	
Conversation with a Computer	\$29.50
KNOW Technologies (Software) Inc.	
SMART DISK	\$39.95
Lattice	
dbc III	\$150.00
Lattice MacLibrary	\$100.00
Lattice Screen Editor	\$100.00
Lattice Text Utilities	\$75.00
Unicalc	\$79.95
Amiga Lattice C Compiler	\$149.95
Amiga MS-DOS C Cross Compiler	\$250.00
Amiga Programmer's Library	\$200.00
Amiga-Lint	\$98.00
dbc III Library	\$150.00
MacLibrary	\$100.00
Make Utility	\$125.00
Panel	\$195.00
Screen Editor	\$100.00
Text Utilities	\$75.00
Manx Software Systems	
Aztec C68k/Am-c	\$499.00
Aztec C68k/Am-d	\$249.00
MacroWare/Kent Engineering & Design	
MacroModem	\$69.95
Mark of the Unicorn	
Hex	\$39.95
Megasoft Limited	
A Copier	\$39.95
A Disk	\$29.95
Meta-Soft Inc.	
C Cross-Reference Utility	\$49.95
LTR	\$59.95
Micro-shell	\$79.95
Metacomco/ Tenchstar Inc.	
Cambridge LISP	\$199.95
ISO Pascal	\$99.95
MCC Utilities	\$49.95
Metadigm Inc.	
MetaScope	\$95.00
Metascope:	
The Debugger	\$95.00
Metascribe: The Editor	\$85.00
Metatools I	\$69.95
MicroDimensions Inc.	
PED	\$29.95
Programmers Toolkit	\$124.95
MicroMaster Software	
Digital Building System	\$299.00
Pick Your Preferences	\$29.95
Microsmith's Inc.	
TxED	\$39.95
TD	\$59.95

Northwest Machine Specialties	
HexDump	\$19.95
Screen Mapper	\$89.95
Omega Star Software	
OnLine Amiga Basic Manual	\$29.95
Organic Productions	
Amiga Training Tapes	\$29.95
Pecan Software Systems	
Power Systems	\$99.95
USCD Pascal	\$79.95
Quelo Inc.	
Cross-Assembler to Amiga External Port (68000)	\$595.00
Cross-Assembler to Amiga External Port(68020)	\$750.00
Cross-Assembler to IBM (68000)	\$199.00
Cross-Assembler to IBM (68020)	\$249.00
Native (68020)	\$149.00
Native(68000)	\$99.00
Source Code (68020)	\$3,500.00
Source Code(68000)	\$3,500.00
Rick Stiles	
UEdit	\$69.95
Softeam Inc.	
PC/ET Emulator	\$69.95
Software Supermarket	
Printer Driver Maker	
Spencer Organization Inc.	
APL 68000	
Syquest	
Quick Test 1000	\$800.00
TDI Software Inc.	
Modula-2	\$89.95
Tecni Soft	
T-Link	\$59.95
T-Move	\$49.95
T-Packs	\$49.95
T-UTIL	\$49.95
The Great American Softworks	
IdeaCraft	\$89.00
The Micro Forge	
Programmer's Editor	\$69.95
Prolog Level 1	\$89.95
Ram Disk	\$24.95
Tigress	
DISKWIK	\$49.95
Transtime Technology	
Datamat A-100	\$125.00
Datamat A-200	\$249.95
Datamat A-300	\$349.95
Datamat EBU	\$40.00
RTD	\$95.00
True BASIC Inc.	
3-D Graphics Library	\$49.95
Advanced String Library	\$49.95
Algebra	\$49.95
Calculus	\$49.95
Chance	\$49.95
Chipendale	\$49.95
Developer's Tool Kit	\$49.95
Discrete Math	\$49.95
Pre-Calculus	\$49.95
Probability	\$49.95
Sorting & Searching	\$49.95
Trigonometry	\$49.95
True Basic Language System	\$149.95
True Stat	\$79.95
Tychon Technologies Inc.	
Tychon Utilities	\$49.95
Canon Printer Driver	\$100.00
UBZ Software	
UBZ Forth	\$85.00
VeraSoft	
dBMan	\$149.95

INTRODUCTORY OFFER
FREE UPDATES

Announcing ...

CHARTMAKER™

the new chart program for the Amiga that allows you to create charts and graphs which use the full power of the Amiga.

- Bar charts, 3D bar charts, line charts, area charts, pie charts, 3D pie charts, and more.
- Options include 3D, patterns, outline, drop shadow, lines, ticks, legends, and more.
- Completely IFF compatible; save your chart, then load it under your favorite paint program and add finishing artistic touches. Use many chart images in a "slide show."
- Spreadsheet style data editor for easy data input.
- Easy to use mouse / menu interface, plus 4096 user selected colors.

Dealer inquiries encouraged.

DISKETTE AND MANUAL \$59.00

send check or money order to:

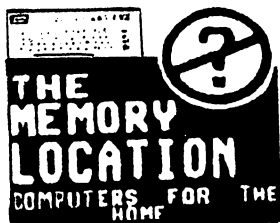
SOUTH PARK SOFTWARE

115 South Park
San Francisco, CA 94107
415 • 957 • 1963

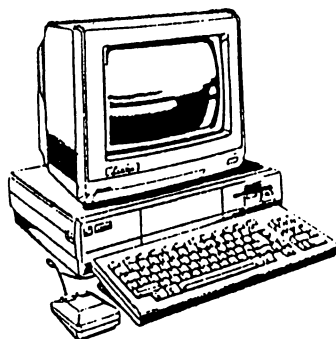
We accept VISA & MasterCard

Chartmaker is a registered trademark of South Park Software.

Amiga is a registered trademark of Commodore-Amiga, Inc.



396 Washington Street
Wellesley, MA 02181
(617) 237-6846



- AMIGA OWNERS -
IMAGINE A STORE BUILT AROUND THE AMIGA!
IT'S HERE NOW!!

THE MEMORY LOCATION
396 WASHINGTON STREET (RT. 16)
WELLESLEY, MA 02181
617 - 237 - 6846

JUST A FEW DOORS UP FROM THE PLAYHOUSE
FEATURING THE LATEST AND THE GREATEST FOR AMIGA
WHAT DO WE HAVE?

A.S.D.G MINI-RACK-B TWO SLOT CARD CAGE +RAM
SUPER HUEY JUMP DISK EXPERT SYSTEM MEAN18
CHESSMASTER 2000 ADDISON WESLEY MANUALS
THE PAWN CRIMSON CROWN TRANSYLVANIA TRINITY
WINNIE THE POOH CHESSMATE CASIO KEYBOARDS
FLIGHT SIMULATOR II INSIDE AMIGA GRAPHICS
ADVANCED BASIC DOS REF. GUIDE TRUE BASIC
THE EXPLORER LEATHER GODDESSES OF PHOBOS
MONEY MENTOR ZUMA FONTS DELUXE HELP GRABBIT
DB-MAN DATAMAT GENESIS II IMPACT GRAPHICS
LOGISTIX LEADER BOARD DELUXE VIDEO DIABLO
FINANCIAL PLUS INFO BASE LATTICE C ZORK I
DELUXE PAINT MASTERTYPE MOUSTERPIECE
FINANCIAL COOKBOOK BRATTACUS HACKER FORTH
SEVEN CITIES OF GOLD ONE ON ONE MARAUDER
TALKING COLORING BOOK ANALYZE! TEXTCRAFT
AEGIS ANIMATOR ZORK II AEGIS IMAGES LISP
MONKEY BUSINESS FORTRAN 77 SPELLBREAKER
AZTEC C SCRIBBLE ZORK III DIGITAL LINK
RACITOR ARCHON GISMOZ CUSTOM PRINT DRIVERS
AMIGA DOS MANUAL (BANTAM) KID TALK BBS-PC
TYCHON UTILITIES PAK-A-DISK MOUSE MATS
ON-LINE AMIGA HANDBOOK (SUNSHINE) FLOW
MOUSTERPIECE HALLEY'S PROJECT PASCAL
GRAPHICRAFT CSI MULTI-FOURTH ARCTIC FOX
PAR-HOME CABLES MINDSHADOW MUSIC STUDIO
BORROWED TIME DISCOVERY AMIGA MODEM T&E
TALKING TRIVIA DIGI-VIEW META-PASCAL
MODULA II DEVELOPERS + COMMER. SPELLER BEE
ELEMENTARY AMIGA BASIC BOOK INFOMINDER
BEGINNERS GUIDE TO AMIGA AMI-PROJECT
AMIGA CROSS DEVELOPMENT ENVIRONMENT FOR IBM
MIND FOREVER VOYAGING BUSINESS STATISTICS
TYPING TUTOR + WORD INVADERS VOLKSMODEM 12
MIAMIGA LEDGER+FILE EXPERIMENTAL STATISTICS
MAXIPLAN SALES FORECASTING VIP PRO. MIRROR
ONE MEG RAM EXPANDER INFOMINDER FISHDISKS
AMICUS DISKS DYNAMIC-CAD GOLDEN HAWK MIDI
MIMETICS SOFTWARE, MIDI, DIGITIZER MAXIPLAN
GOLDEN OLDIES OKIMATE 20 PRINTER
AMAZING COMPUTING AMIGA WORLD TRANSACTOR
CANON COLOR INK JET AND DRIVER
SOFTWARE RENTAL CLUB CONSIGNMENT SALES

AND MORE !!!

A BETTER QUESTION WOULD BE
"WHAT DON'T WE HAVE?"

ONLY WHAT WORKS, SATISFACTION GUARANTEED

Westcomp	
Amiga Intelligence	\$149.95
CHEM 101	\$39.95
Co-Pilot	\$49.95
Pilot interpreter	\$30.00
The Vise	\$150.00

Zoxso	
Imagine This	
ZLI	\$49.95

Miscellaneous Software

Associated Computer Services	
Grade Manager	\$89.95
Paraclete	\$89.95
Quiz Master	\$79.95
Station Manager	\$1,995.00
Emusoft Corp	
ESSC Calculate	

Music

Commodore	
Musicraft	\$99.95
Electronic Arts	
Instant Music	\$49.95
Instant Music Data Disk	
Magnetic Music	
Texture	\$350.00
Micro W	
QRS Music Rolls	\$19.95
Mimetics	
DX Connect	\$149.00
RX Connect	\$149.00
SMPTE Time Code	
SoundScape	\$149.00
SoundScape PRO MIDI Studio	\$149.00
SX Connect	\$149.00
Revolution Software	
BobShop/Soundshop	\$49.95
Speech System	
Casio Connection	
Symphony Music Libraries	\$39.95
Symphony Writer	
The MIDI Symphony	\$99.95
Sterling Software	
DNA Music	\$19.95

Telecommunications

Benaiah Computer Products, Inc.	
Middleman	\$59.95
Commodore Business Machines	
Amiga Term	
Developers of Advanced Software	
D.A.S. Communications	\$30.00
Eight Stars Software	
Starnet	\$99.95
Felsina Software	
A-Talk	\$49.95
Megasoft Limited	
A Term	\$49.95
Micro-Systems Software	
BBS-PC	\$99.95
On-Line	\$69.95
SKE Software Co.	
SKETerm	\$49.95

Books

Addison-Wesley Publishing	
Amiga Hardware Reference Manual	\$24.95
Amiga Intuition Kernel Reference Manual	\$24.95
Amiga ROM Kernel Reference Manual	\$24.95
Bantam Electronic Publishing	
The Amiga DOS Manual	\$24.95
Compute Books	
Advanced Amiga BASIC	\$16.95
COMPUTE's Amiga Applications	\$16.95
COMPUTE's Amiga Programmer's Guide	\$16.95
COMPUTE's AmigaDOS Reference Guide	\$14.95
COMPUTE's Beginner's Guide to the Amiga	\$16.95
COMPUTE's Kids and the Amiga	\$14.95
Elementary Amiga Basic	\$14.95
Inside Amiga Graphics	\$16.95

Hardware

Akron Systems Development	
A-Time Real Clock Calendar	\$59.95
Anakin Research, Inc.	
Easyl	\$499.00
Anchor Automation	
Vols Omega 80	\$199.00
Applied Visions	
FutureSound	\$175.00
Aprotek	
Amiga Parallel Printer Cables	\$21.95
ASDG Incorporated	
1Mb FAST Ram Board	\$650.00
2Mb FAST Ram Board	\$900.00
512k FAST Ram Board	\$450.00
Mini-Rack-C	\$195.00
Mini-Rack-D	\$325.00
Belkin Components	
Amiga Parallel Printer Cables	\$18.00
Four-Way Data Transfer Switch	\$129.00
Two-Way Parallel Data Switch	\$99.00
Burklund & Associates	
GenLock Subsystem, Model RM2	\$850.00
Byte by Byte	
The Pal	
Commdore-Amiga	
Genlock 1300	
LIVE!	
Component Systems, Inc.	
Multiport Controller	\$877.00
Comspec Communications	
Amiga 2 meg RAM Board	\$1,079.00
Cypress Technologies, Inc.	
2MB Ram Expansion	
Digital Systems Engineering	
Desktop_Amp	\$44.95
Disk_Mate	\$89.95
DSI	
Buss Station	
Duryea Associates, Inc.	
MIDAS	\$15,000.00
Golden Hawk Technology	
MIDI GOLD	\$79.00
Hippopotamus Software, Inc.	
Hippo Clean	\$29.95
Home Controller	\$139.95
Sound Digitizer	\$199.95
Interactive Video Systems	
IVS Busbox	\$299.95
IVS Magnus	\$99.95
IVS Ramex-1M	\$199.95

Amiga C. Itoh 8510 ProDriver

Color **Plus** Color

\$44.95* Software only

\$64.95* Software & cable

Supports Printers:

- 8510 A, AP+, B, S, S+, SC (color) & SC+ (color)

Works with:

- Textcraft¹, Scribble!², Deluxe Paint³ & others

MicroCybernetics Corporation

P.O. Box 3126

Laurel, Maryland 20708

(301) 498-5704

Dealer inquiries accepted

*Maryland residents enclose 5% sales tax

Trademarks of: Commodore Amiga¹, Micro System Software² & Electronic Arts³



MOUSE TRAP

STATIC TRAPPING MOUSE PAD

What do you get when you combine the optimum surface for mouse to run on, and static protection for your entire computer system?

MOUSE TRAP, the static trapping mouse pad!

Crystal Computer has produced the complete answer for the computer mouse user. Keep your mouse clean and protect your system from the devastating effects of static discharge. MOUSE TRAP is designed to give your mouse a smooth surface to run on, while maintaining the traction needed for the mouse ball. The surface is cleanable, even with industrial solvents and is five times harder than convention desk tops. It's 8 1/2 X 11 inch size fits your work station, and because it's made of foam you can write over it when your mouse is not in use.

MOUSE TRAP is only \$49.95 including shipping.

Call or write,



Crystal
Computer Inc.

2286 E. Steel Road St. Johns, MI 48879
Ph. 1-800-245-7316 24 hrs. 7 days
Ph. 1-517-224-7667 in Michigan

ATTN:
PASCAL
USERS

MODULA-2

the successor to Pascal

- FULL interface to ROM Kernel, Intuition, Workbench and AmigaDos
- Smart linker for greatly reduced code size
- True native code implementation (Not UCSD p-Code or M-code)
- Sophisticated multi-pass compiler allows forward references and code optimization
- RealInOut, LongInOut, InOut, Strings, Storage, Terminal
- Streams, MathLib0 and all standard modules
- Works with single floppy/512K RAM
- Supports real numbers and transcendental functions ie. sin, cos, tan, arctan, exp, ln, log, power, sqrt
- 3d graphics and multi-tasking demos
- CODE statement for assembly code
- Error lister will locate and identify all errors in source code
- Single character I/O supported
- No royalties or copy protection
- Phone and network customer support provided
- 350-page manual

Pascal and Modula-2 source code are nearly identical. Modula-2 should be thought of as an enhanced superset of Pascal. Professor Niklaus Wirth (the creator of Pascal) designed Modula-2 to replace Pascal.

Added features of Modula-2 not found in Pascal

- CASE has an ELSE and may contain subranges
- Programs may be broken up into Modules for separate compilation
- Machine level interface
 - Bit-wise operators
 - Direct port and Memory access
 - Absolute addressing
 - Interrupt structure
- Dynamic strings that may be any size
- Multi-tasking is supported
- Procedure variables
- Module version control
- Programmer definable scope of objects
- Open array parameters (VAR r: ARRAY OF REALS;)
- Elegant type transfer functions

Ramdisk Benchmarks (secs)	Compile	Link	Execute	Optimized Size
Sieve of Eratosthenes:	6.1	4.9	4.2	1257 bytes
Float	6.7	7.2	8.6	3944 bytes
Calc	5.7	4.8	3.6	1736 bytes
Null program	4.8	4.7	—	1100 bytes

```

MODULE Sieve;
CONST
  Size = 8190;
TYPE
  FlagRange = [0..Size];
VAR
  FlagSet: SET OF FlagRange;
  i: FlagRange;
  Prime, k, Count, Iter: CARDINAL;
BEGIN
  ('SS-SR-SA*')
  FOR Iter:= 1 TO 10 DO
    Count:= 0;
    Flags:= FlagSet(); (* empty set *)
    FOR i:= 0 TO Size DO
      IF (i IN Flags) THEN
        Prime:= (i * 2) + 3; k:= i + Prime;
        WHILE k <= Size DO
          INCL (Flags, k);
          k:= k + Prime;
        END;
        Count:= Count + 1;
      END;
    END;
  END;
END;
END Sieve.

```

```

MODULE Float;
FROM MathLib0 IMPORT sin, ln, exp,
                                sqrt, arctan;
VAR x,y: REAL; i: CARDINAL;
BEGIN (*ST-SA-SS-*)
  x:= 1.0;
  FOR i:= 1 TO 1000 DO
    y:= sin (x); y:= ln (x); y:= exp (x);
    y:= sqrt (x); y:= arctan (x);
    x:= x + 0.01;
  END;
END float.

```

```

MODULE calc;
VAR a,b,c: REAL; n, i: CARDINAL;
BEGIN (*ST-SA-SS-*)
  n:= 5000;
  a:= 2.71828; b:= 3.14159; c:= 1.0;
  FOR i:= 1 TO n DO
    c:= c*a; c:= c*b; c:= c/a; c:= c/b;
  END;
END calc.

```

Product History

The TDI Modula-2 compiler has been running on the Pinnacle supermicro (Aug. '84), Atari ST (Aug. '85) and will soon appear on the Macintosh and UNIX in the 4th Qtr. '86.

Regular Version \$89.95 Developer's Version \$149.95 Commercial Version \$299.95

The regular version contains all the features listed above. The developer's version contains additional Amiga modules, macros and demonstration programs – a symbol file decoder – link and load file disassemblers – a source file cross referencer – the kermit file transfer utility – a Modula-2 CLI – modules for IFF and ILBM. The commercial version contains all of the Amiga module source files.

Other Modula-2 Products

Kermit – Contains full source plus \$15 connect time to Compuserve. \$29.95
 Examples – Many of the C programs from ROM Kernel and Intuition translated into Modula-2. \$24.95
 GRID – Sophisticated multi-key file access method with over 30 procedures to access variable length records. \$49.95



SOFTWARE, INC.

10410 Markison Road • Dallas, Texas 75238 • (214) 340-4942
 Telex: 888442 Compuserve/Number: 75026.1331

Johnathon Freeman Designs

Converter + \$279.95
 Universal Interface Converter \$99.00
 Universal Printer/Plotter Buffer \$329.00

Kurta Corp.

Penmouse+ \$295.00
 Series One \$695.00

Media Technology Associates

MTA 200 \$79.95
 MTA Series 1000

MEGAmiga

1Mb RAM Board \$512.00
 2Mb RAM Board \$768.00

MicroBotics, Inc.

Scuzzy 20MB Hard Disk \$1,495.00
 256K Memory Expansion \$149.95
 2MB Memory Expansion \$495.00

Newtek

Digi-View \$199.00

Okidata

Okimate 20 \$268.00

Remote Measurement Systems, Inc.

ADC-1 Data Acquisition And Control System \$449.00

RS Data Systems

Pow-R-Card

Skyles Electric Works

256K Memory Expansion \$149.95
 Megabytes
 MIDI For Amiga \$79.95
 Clock For Amiga \$59.95

STACAR International

Disk Drive Expansion Box \$225.00

Starpoint Software

256K Memory Expansion \$120.00

Tecmar

T-Disk

The Gemstone Group

Amiga Expansion Box \$995.00

The Micro Forge

7 Slot Expansion Box \$84.95
 Memory/Hard Disk Expansion \$656.95
 Stereo Sound Digitizer \$344.95
 Expansion Box \$656.95
 Hard Drive 10 meg \$1,415.95
 Hard Drive 20 meg \$1,494.95
 Hard Drive 30 meg \$1,762.95

Miscellaneous

T's Mee

Amiga Sweatshirts \$15.00
 Amiga Tee-Shirts \$9.50

Crystal Computer, Inc.

Mouse Trap (anti-static) \$49.95

Important Note:

This list was composed over time and may be missing items from new suppliers. If you are such a supplier and you are not represented or the entry is incorrect, please let us know.

Send all Press releases to:

Amazing Directory
 Amazing Computing
 PIM Publications, Inc.
 P.O. Box 869
 Fall River, MA. 02722

Amiga Developers

Absoft Corp. 4268 N. Woodward Royal Oak MI 48072 (313) 549 7111	B.E.S.T. Inc. P.O. Box 852 McMinnville OR 97128 (503) 472 9512	Colony Software 931 W 21st Street Norfolk VA 23517 (804) 625-1945	DeluxeHelp Inc. Box 249, 4356 Okeechobee Blvd. W.Palm Beach FL 33409 (305) 622 0138	Ensign Software 7337 Northview Boise ID 83704 (208) 378 8086
Academy Software, Inc. PO Box 6277 San Rafael CA 94903 (415) 499 0850	Bantam Electronic Publishing 666 5th Ave. New York NY 10103 (212) 765 6500 Batteries Included 30 Mural St. Richmond Hill Ontario Canada L4B 1B5 (416) 881 9941	Commadore Business Machines 1200 Wilson Dr. West Chester PA 19380 (215) 431-9100	Desktop A.I. 1720 Post Road E. Westport CT 06880 (203) 255 3400	Epyx Inc. 1043 Kiel Court Sunnyvale CA 94089 (408) 745 0700
Access Software 2561 S. 1560 W. Woods Cross UT 84087 (801) 298 9077	Baudville 1001 Medical Park Dr. SE Grand Rapids MI 49506 (616) 957 3036	Component Systems, Inc. 778-A Brannan St. San Francisco CA 94103 (415) 861 1345	Deskware PO Box 47577 St. Petersburg FL 33743	Felsina Software 3175 S. Hoover St., Suite 275 Los Angeles CA 90007 (213) 747 8498
Accolade 20823 StevensCreek Blvd.C1A Cupertino CA 95014 (408) 446 5757	Belkin Components 4718 W. Rosecranss Ave. Hawthorne CA 90250 (800) 223 5546	Compumed PO Box 6939 Salinas CA 93912 (408) 758 2436	Developers of Advanced Software 12455 Veterans Memorial Dr., Ste. 204 Houston TX 77014	Film Production Toolkit 446 Sherman Canal Ct. Venice CA 90291 (213) 306 2146
Activision 2350 Bayshore Mountain View CA 94043 (415) 960 0410	Benaliah Computer Products, PO Box 11165 Huntsville AL 35814 (205) 859 9487	Compute Books Box 5038, FDR Station New York NY 10150 (800) 346-6767	Digital Creations 530 Bercut Drive Suite F Sacramento CA 95814 (916) 446 4825	Finally Software 4000 MacArthur Blvd. Newport Beach CA 92663 (714) 854 4434
Addison-Wesley Publishing Jacob Way Reading MA 01867 (617) 944 3700	Bethesda Softworks 9208 Burning Tree Rd. Bethesda MD 20817 (301) 469 7061	Computer Food Inc. 2215 Sarah Court, Suite 80H Norcross GA 30093 (404) 851 9103	Digital Solutions 30 Wertheim Court, #2 Richmand Hill, Ontario Canada L4B 1B9 (416) 731 8775	Firebird Software PO Box 49 Ramsey NJ 07446 (201) 934 7373
Aegis Development 2210 Wilshire Blvd. Santa Monica CA 94043 (213) 306 0735	Burklund & Associates 3903 Carolyn Ave. Fairfax VA 22031 (703) 273 5663	Computer Solutions PO Box 354, 888 S. Elfert Mason MI 48854 (800) 874 9375	Digital Systems Engineering Blowing Wind Way Citrus Heights CA 95621 (916) 725 3025	First Byte 2845 Temple Ave. Long Beach CA 90806 (213) 595 7006
Akron Systems Development PO Box 6408 Beaumont TX 77705 (409) 833-2686	Byte by Byte 3736 Bee Cave Rd., Suite 3 Austin TX 78746 (512) 328 2983	Comspec Communications 153 Bridgeland Ave. Toronto, Ontario Canada M6A 2Y6 (416) 787-0617	Digitek Software Development 2519 Bonita Dr. Highland CA 92346 (714) 864-2846	Foxware 1554 Park Creek Ln. Atlanta GA 30319
Alive Systems PO Box 50 Big Sur CA 93920 (415) 332 8018	Capilano Computer Systems PO Box 86971 N. Vancouver, BC, Canada V7L 4P6 (604) 669 6343	Creative Solutions 4701 Randolph Rd. Suite 12 Rockville MD 20852 (301) 984 0262	Discovery Software 262 S. 15 St., Suite 300 Philadelphia PA 19102 (215) 546 1533	Gander Software 3223 Bross Rd., The Ponds Hastings MI 49058 (616) 945 2821
Amiga Man PO Box 58768 Houston TX 77258 (713) 480 1735	CBS Interactive Learning 1 Faucett Place Greenwich CT 06836 (203) 622 2500	Crystal Computer, Inc. 2286 E. Steel Road St. Johns MI 48879 (800) 245-7316	DSI 717 South Emporia Wichita KS 67211 (316) 264 6118	Geodesic Publications PO Box 7 Willow Creek CA 95573 (916) 629 3514
Anakin Reasearch, Inc. 100 Westmore Dr., Unit 11C Rexdale, Ontario Canada M9V 5C3 (416) 744 4246	Central Coast Software 268 Bowie Drive Los Osos CA 93402 (805) 528-4906	Crystal Rose Software 109 S. Los Robles Pasadena CA 91101 (818) 795 6664	Duryea Associates, Inc. 701 Alpha Rd. Pittsburg PA 15238 (412) 963 7262	Gimpel Software 3207 Hogarth Lane Collegeville PA 19426 (215) 584 4261
Anchor Automation 6913 Valjean Ave. Van Nuys CA 91406 (818) 997 6493	Chang Labs 5300 Stevens Creek Blvd. San Jose CA 95129 (408) 246 8020	Cypress Technologies, Inc. PO Box 3346 Fremont CA 94539 (415) 656 1974	Eastern Telecom, Inc. 9514 Brinton Drive Orlando FL 32817 (305) 657-4355	Golden Hawk Technology 427-3 Amherst Street Nashua NH 03063 (603) 882-7198
Applied Visions 15 Oak Ridge Rd. Medford MA 02155 (617) 488 3602	Classic Image 510 Rhode Island Ave. Cherry Hill NJ 08002 (609) 667 2526	Dark Horse Dept. A9, Box 36162 Greensboro NC 27416 (919) 852 3698	Eclipse Data Management 312.5 Lafayette St. Glendale CA 91205 (818) 956 5517	Grey Associates 250 Bruton Way Atlanta GA 30342 (404) 851 9103
Aprotek 1071-A Avenida Acaso Camarillo CA 93010 (805) 987 2454	Classic Image Software 510 Rhode Island Ave. Cherry Hill NJ 08002 (609) 667 2526	Data Dynamics, Inc. PO Box 2728 Portland OR 97208 (503) 626 4635	Eight Stars Software 2900 Boniface Pkwy. Anchorage AK 99504 (907) 338 1246	Harvsoft Box 725 Kennmore NY 14217 (716) 877 3510
ASDG Incorporated River Road Suite #54A Piscataway NJ 08854 (201) 540-9670	Clockwork Computers 2215 Sarah Court, Suite 80 Norcross GA 30093 (404) 851 9103	Data Research Processing 5121 Audrey Dr. Huntington Beach CA 92649 (714) 840 7186	Electronic Arts 1820 Gateway Dr. San Mateo CA 94404 (415) 571 7171	HC Software Australia GPO Box 2204 Adelaide South Australia 5001; 08-428377
Associated Computer Services 839 S. Glenstone Springfield MO 65802 (417) 865 6555		Delta Research 9054 Wilkie Way Palo Alto CA 94306 (415) 856 3669	Emusoft Corp. 1400 Chicago Ave., Ste. 303 Evanston IL 60201 (312) 869 6676	

Infocom 125 Cambridge Park Drive Cambridge MA 02140 (617) 492 1031	Marksman Tech Inc. Route 5, Box 221A Santa Fe NM 87501 (505) 455 2681	Microsmith's Inc. P.O. Box 561 Cambridge MA 02140 617 576 2878	Queue Intellectual Software 5 Chapel Hill Drive Fairfield CT 06432 (203) 335 0908
Insight/Lehner Communications 2708 Arlington Highland Park IL 60035 (312) 432 5458	Maxisoft 2812 Sloat Rd. Pebble Beach CA 93953 (408) 825 4104	Mimetics PO Box 60238 Sta. A Palo Alto CA 94306 (408) 741 0117	Quicksilver Software 418 W. 7th St. Sioux City IA 51103 (712) 258 2018
Interactive Analytic Node 2345 West Medicine Lake Dr. Minneapolis MN 55441 (612) 871 6283	Media Technology Associates 9208 Burning Tree Rd. Bethesda MD 20817 (301) 469 7060	Mindscape 3444 Dundee Road Northbrook IL 60062 (312) 480 7667	Ramella 1493 Mountain View Ave. Chico CA 95926
Interactive Microsystems Box 272 Boxford MA 01921 (617) 887 9607	MEGAmiga 1620 N. Park Ave. Tucson AZ 85719 (602)-622-1344	New Horizons Software PO Box 180253 Austin TX 78718 (512) 280 0319	Remote Measurement Systems, Inc. 2633 Eastlake Ave. E, Ste. 206 Seattle WA 98102 (206) 328 2255
Interactive Video Systems 15201 Santa Gertrudes Ave., Y-102 La Mirada CA 90638 (714) 739 5020	MegaSoft Limited PO Box 1080 Battle Ground WA 98604 (206) 687 5205	Newtek 701 Jackson Suite B3 Topeka KS 66603 (913) 354-9332	Revolution Software PO Box 38 West Chester PA 19381 (215) 430 0412
JDK Images 2224 E. 86th St., Suite 14 Bloomington MN 55420 (612) 854 7793	Megatronics 55 N. Main St. Logan UT 84321 (801) 752 2642	Nimbus P.O. Box 1433 Ranch Santa Fe CA 92067 (619) 944-3453	Rick Stiles 2420 Summit Springs Dr. Dunwoody GA 30338 (404) 587 5396
Jenday Software P.O. Box 4313 Garden Grove CA 92642 (714) 636 3378	Meridian Software PO Box 890408 Houston TX 77289-0408 (713) 488 2144	Northeast Software Group 165 Dyerville Ave. Johnston RI 02919 (401) 273 1001	Rosetta-Stone Software 4000 MacArthur Blvd., Ste. 3000 Newport Beach CA 92663 (714) 854 4434
JMH Software of Minnesota 7200 Hemlock Lane Maple Grove MN 55369 (612) 424 5464	Meta-Soft Inc. PO Box 7293 Las Cruces NM 88006 (505) 523 0371	Northwest Machine Specialties 3611 Joshua NE Salem OR 97305 (503) 588 0008	RS Data Systems 7322 Southwest Freeway, Ste. 660 Houston TX 77074 (713) 988 5441
Johnathon Freeman Designs 1067 Dolores St. San Francisco CA 94110 (415) 822 8451	Metacomco/ Tenchstar Inc. 201 Hoffman Ave. Monterey CA 93940 (408) 375 5012	Okidata 532 Fellowship Rd. Mount Laurel NJ 08054 (800) 654 3282	RTL Programming Aids 10844 Deerwood SE Lowell MI 49331 (616) 897 5672
KNOW Technologies (Software) Inc. 9651 Alexandria Road Richmond, BC Canada V6X 1C6 (604) 270-0064	Metadigm Inc. 19762 MacArthur Blvd. Suite 300 Irvine CA 92715 (714) 955 2555	Olamic Systems Corporation 141 West Jackson Blvd. Chicago IL 60604 (312) 786 1410	S. Anthony Studios 889 De Haro St. San Francisco CA 94107 (415) 826 6193
Kurta Corp. 4610 S. 35th St. Phoenix AZ 85040 (602) 276 5533	Micro Prose 120 Lakefront Dr. Hunt Valley MD 21030 (301) 667 1151	Omega Star Software PO Box 1831 Clemson SC 29633 (803) 882 3608	Scarborough Systems Inc. 55 S. Broadway Tarrytown NY 10591 (914) 332 4545
Lattice 22 W. 600 Butterfield Rd. Glen Ellyn IL 60137 (312) 858 7950	Micro W PO Box 198 Butler NJ 07405 (201) 838 5608	Organic Productions 71 Gold St. East Hartford CT 06118 (203) 569 3855	Scott Lamb 205C Heights Ln. Ft. Worth TX 76112 (817) 496 9220
Lionheart Business Software P.O. Box 329 Albany VT 05440 (514) 933 4918	Micro-Systems Software Inc. 4031 Oak Circle Boca Raton FL 33431 (800) 327 8724	PAR Software PO Box 1089 Vancouver WA 98666 (206) 695 1368	Sedonna Software 11844 Rancho Bernardo Rd. San Diego CA 92128 (619) 451 0151
MacroWare/Kent Engineering & Design 3208 Bettline Rd., Suite 210 Mottville NY 75234 (315) 685-8237	MicroBotics, Inc. PO Box 855115 Richardson TX 75085 (214) 437 5330	Pecan Software Systems 1410 39th Street Brooklyn NY 11218 (718) 851 3100	Sierra On-Line Inc. Box 485 Coarsegold CA 93614 (209) 683 6858
Magnetic Music PO Box 328 Rhinebeck NY 12572 (914) 876 4845	MicroDimensions Inc. 455 N. University Ave. Provo UT 84601 (801) 377 0933	Polarware PO Box 311 2600 Keslinger Rd. Geneva IL 60134 (312) 232 1984	SKE Software Co. 2780 Cottonwood Court Clearwater FL 33519 (813) 786 3247
Manx Software Systems One Industrial Way Eatontown NJ 07724 (800) 221 0440	MicroEd Inc. PO Box 444005 Eden Prairie MN 55344 (612) 944 8740	Progressive Peripherals and Software 464 Calamath St. Denver CO 80204 (303) 825 4144	Skyles Electric Works 231-E South Whisman Rd. Mountain View CA 94041 (800) 227 9998
Manx Software Systems Inc. 208 Maple Avenue Red Bank NJ 07701 (201) 530 7997	Microillusions PO Box 3475 Granada Hills CA 91344 (818) 360 3715	Quso Inc. 2464 33rd Ave. W, Suite 173 Seattle WA 98199 (206) 285 2528	Soft Circuits Inc. 401 S.W. 75th Terrace N. Lauderdale FL 33068 (305) 721 2707
Mark of the Unicorn 222 Third St. Cambridge MA 02142 (617) 576 2760	MicroMaster Software 1289 Broadhead Rd. Monaca PA 15061 (412) 775 3000	Queue Inc. 798 North Avenue Bridgeport CT 06606 (203) 333 7268	Softeam Inc. 14420 Harris Place Miami Lakes FL 33014 (305) 825 4820

Software Group/Amiga Northway Ten Executive Park Ballston Lake NY 12019 (518) 877-8600	SubLOGIC 713 Edgebrook Dr. Champlagn IL 61820 (217) 359 8482	The Micro Forge 398 Grant St. SE Atlanta GA 30312 (404) 688 9464	UBZ Software 395 St. Albans Court Mableton GA 30059
Softwood Co. PO Box 2280 Santa Barbara CA 93120 (805) 966 3252	Syquest PO Box 758 Snowdon Station, Montreal, Quebec Canada H3X 3X9 (514) 935 5881	The Other Guys 55 N. Main St., Suite 301D, PO Box H Logan UT 84321 (800) 942 9402	Unicorn Software Co. 2950 E. Flamingo Rd. Las Vegas NV 89121 (702) 732 8862
Softworks 2944 N. Broadway Chicago IL 60657 (312) 975 4030	T.R. Software 4346 W. Maypole Chicago IL 60624 (312) 875 9760	The Other Valley Software 8540 Archibald Suite A Rancho Cucamonga CA 91730 (714) 980 0440	VeraSoft 723 Seawood Way San Jose CA 95120 (408) 268 6033
Speech Systems 38W255 Dearpath Road Batavia IL 60510 (312) 879-6680	TDI Software Inc. 10410 Markison Rd. Dallas TX 75238 (214) 340 4942	The Quality Cottage 6301 F University Commons South Bend IN 46635 (219) 234 4401	VIP Technologies 132 Aero Camino Santa Barbara CA 93117 (805) 968 9567
Spencer Organization Inc. 366 Kinderkamak Rd. Westwood NJ 07675 (201) 666 6011	Tecni Soft PO Box 7175, 5505 Walden Maedows Rd. Murray UT 84123 (801) 268 4961	Tigress PO Box 665 Glendora CA 91740 (818) 334 0709	Westcomp 517 N. Mountain Ave.Sts. 229 Upland CA 91786 (714) 982 1738
STACAR International 14755 Ventura Blvd., Suite 1-812 Sherman Oaks CA 91403 (818) 904-1262	The Computer Club 4843A S. 28th St. Arlington VA 22206 (703) 998 7588	Transtime Technology 797 Sheridan Dr. Tonawanda NY 14150 (716) 874 2010	WordPerfect Corp. 288 W. Center St. Orem UT 84057 (801) 227 4000
Starpoint Software 122 S. Broadway Yreka CA 96097 (916) 842 6183	The Dragon Group 148 Poca Fork Rd. Elkview WV 25071 (304) 965 5517	True BASIC Inc. 39 S. Main St. Hanover NH 03755 (603) 643 3882	X-Scope Enterprises PO Box 210063 Columbia SC 29221 (803) 779 0619
Sterling Software 77 Mead St. Bridgeport CT 06610 (203) 366 7775	The Gemstone Group 620 Indian Spring Ln. Buffalo Grove IL 60089 (312) 537 0544	Tychon Technologies Inc. 25000 Euclid Ave. Cleveland OH 44117 (216) 261 7088	Zoxso PO Box 283 Lowell MA 01853 (617) 655 9548
Strategic Simulations 1046 N. Rendstorff Ave. Mountain View CA 94043 (415) 964 1353	The Great American Softworks PO Box 819 Larkspur CA 94939	UBZ Software 395 St. Albans Court Mableton GA 30059 (404) 948 4654	Zuma Group, Inc. 6733 N. Black Canyon Phoenix AZ 85015 (602) 246 4238

•AC•

8 MEGABYTES

Now RS DATA's New POW•R•CARD

Let's You Play Like The Big Boys.

Playing games on your Amiga can be a great deal of fun, but let's be honest — there's more to life than playing games. Now you can turn your computer into a real-life professional machine with the **POW•R•CARD** from **RS DATA** Systems.

The **POW•R•CARD** is a powerful new expansion board which allows you to mature in your computer use with greater flexibility in multi-processing and multi-tasking.

POW•R•CARD starts you off with a 2 Meg capability and allows you to grow with upgrades to a huge 8 Meg RAM expansion, all on the same board so you don't waste valuable slot space. That means you can run more software without fear of Guru Meditation Numbers, out-of-memory crashes or any other small system

boo-boos! What's more, you won't have to rob your piggy bank because **POW•R•CARD** offers this tremendous growth at a cost lower per megabyte than you'll find anywhere.

With your new **POW•R•CARD**, memory expansion is as easy as 1-2-3. The **POW•R•CARD** and enclosure will pass through the Buss without modification for even greater expansion. So you don't have to play games with your data anymore. Graduate to bigger and better things with the **POW•R•CARD** from **RS DATA**!

Upcoming Products from **RS DATA**:

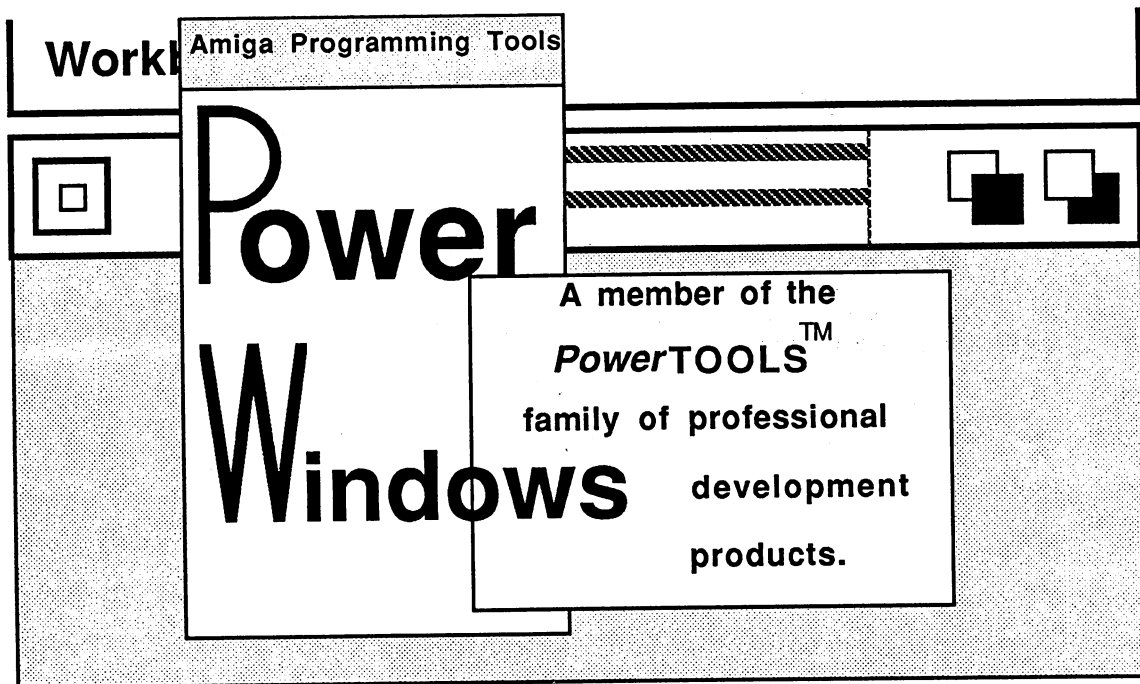
- New Hard Disk System, 20 & 40 megabyte memory.
- 4 Port Parallel card.

- 4 Port Serial Card, allowing more serial type peripheral use.
- 4 Slot Expansion System with horizontal board placement for system height reduction.
- Much, much more!!!

The **POW•R•CARD** is available now from your local Amiga dealer... or call **RS DATA** today!



7322 Southwest Freeway
Suite 660
Houston, Texas 77074
713/988-5441



The **first** interactive Amiga program design tool, *PowerWindows*™ lets you design fantastic looking windows, menus and gadgets in minutes instead of hours or days! You show this incredible program what you want and it does the rest, generating C or 68000 assembler source code for you to include in your own programs. *PowerWindows* is a structure generator for a machine that **thrives** on structures. With this software package, written in assembler, you can:

Pick the exact size and position for your windows **visually**. No more "wait to see what it looks like"; *PowerWindows* knows where your window is and everything else about it!

Design professional looking menus. Add menus, move menus, or delete menus, whatever you want to do with text menus, our program keeps track of them and writes source code letting you duplicate them exactly with simple operating system calls.

Create your own string, integer and boolean gadgets and position them anywhere in your window. *PowerWindows* keeps them from colliding and remembers the type, location and text contents of each one for writing those complex gadget structures.

Best of all, you can keep your designs in a format that can be re-edited, letting you create your favorite type of windows and customize them for each program you write.

Order Form

Price for *PowerWindows* is \$89.95, plus \$3.50 for shipping and handling. Texas residents please add 6.125% sales tax to total price.

Name _____
 Address _____
 City _____ State _____ Zip _____
 Products ordered _____
 Payment method: ☐ MC/Visa ☐ Check ☐ Money Order
 Card Number _____ Expiration Date _____
 Name on card _____ Signature _____
 Enter total enclosed: _____

AC

INOVATRONICS, INC.

11311 Stemmons Frwy., Suite 7

Dallas, TX 75229

214/241-9515

The AMICUS & Fred Fish Public Domain Software Library

This software is collected from user groups and electronic bulletin boards around the nation. Each Amicus disk is nearly full, and is fully accessible from the Workbench. If source code is provided for any program, then the executable version is also present. This means that you don't need the C compiler to run these programs. An exception is granted for those programs only of use to people who own a C compiler.

The Fred Fish disk are collected by Mr. Fred Fish, a good and active friend of the Amiga.

Note: Each description line below may include something like 'S-O-E-D', which stands for 'source, object file, executable and documentation'. Any combination of these letters indicates what forms of the program are present. Basic programs are presented entirely in source code format.

AMICUS Disk 1

ABasic programs: Graphics

3DSolids 3d solids modeling program w/sample data files
Blocks draws blocks
Cubes draws cubes
Durer draws pictures in the style of Durer
FScape draws fractal landscapes
Hidden 3D drawing program, w/ hidden line removal
JPad simple paint program
Optical draw several optical illusions
PaintBox simple paint program
Shuttle draws the Shuttle in 3d wireframe
SpaceArt graphics demo
Speaker speech utility
Sphere draws spheres
Spiral draws color spirals
ThreeDee 3d function plots
Topography artificial topography
Wheels draws circle graphics
Xenos draws fractal planet landscapes

ABasic programs: Tools

AddressBook simple database program for addresses
CardFile simple card file database program
Demo multiwindow demo
KeyCodes shows keycodes for a key you press
Menu run many ABasic programs from a menu
MoreColors way to get more colors on the screen at once, using aliasing
shapes simple color shape designer Speakit speech and narrator demo

ABasic programs: Games

BrickOut classic computer brick wall game
Othello also known as 'go'
Saucer simple shoot-em-up game
Spelling simple talking spelling game
ToyBox selectable graphics demo

ABasic programs: Sounds

Entertainer plays that tune
HAL9000 pretends it's a real computer
Police simple police siren sound
SugarPlum plays "The Dance of the Sugarplum Fairies"

C programs:

Aterm simple terminal program, S-E
cc aid to compiling with Lattice C
decvnt opposite of CONVERT for cross developers
Dotty source code to the 'dotty' window demo
echox unix-style filename expansion, partial S, O-D
fasterfp explains use of fast-floating point math
FixDate fixes future dates on all files on a disk, S-E
freedraw simple Workbench drawing program, S-E
GfxMem graphic memory usage indicator, S-E
Grep searches for a given string in a file, with documentation
ham shows off the hold-and-modify method of color generation
IBM2Amiga fast parallel cable transfers between an IBM and an Amiga
Mandel Mandelbrot set program, S-E
moire patterned graphic demo, S-E
objfix makes Lattice C object file symbols visible to Wack, S-E
quick quick sort strings routine
raw example sample window I/O
setlace turns on interlace mode, S-E

sparks qix-type graphic demo, S-E

Other executable programs:

SpeechToy speech demonstration
WhichFont displays all available fonts
Texts:
68020 describes 68020 speedup board from CSA
Aliases explains uses of the ASSIGN command
Bugs known bug list in Lattice C 3.02
CLICard reference card for AmigaDOS CLI
CLICCommands guide to using the CLI
Commands shorter guide to AmigaDOS CLI commands
EdCommands guide to the ED editor
Filenames AmigaDOS filename wildcard conventions
HalfBright explains rare graphics chips that can do more colors
ModemPins description of the serial port pinout
RAMdisks tips on setting up your RAM: disk
ROMWack tips on using ROMWack
Sounds explanation of the Instrument demo sound file format
Speed refutation of the Amiga's CPU and custom chip speed tips on using Wack

WackCmnds

AMICUS Disk 2

C programs:

alib AmigaDOS object library manager, S-E
ar text file archive program, S-E
fixobj auto-chops executable files
shell simple CLI shell, S-E
sq, usq file compression programs, S-E
YachtC a familiar game, S-E
Make a simple 'make' programming utility, S-E
Emacs an early version of the Amiga text editor, S-E-D

Assembler programs:

bsearch.asm binary search code
qsort.asm Unix compatible qsort() function, source and C test program
setjmp.asm setjmp() code for Lattice 3.02
SVprintf Unix system V compatible printf() function
trees.o Unix compatible tree() function, O-D
(This disk formerly had IFF specification files and examples. Since this spec is constantly updated, the IFF spec files have been moved to their own disk in the AMICUS collection. They are not here.)

John Draper Amiga Tutorials:

Animate describes animation algorithms
Gadgets tutorial on gadgets
Menus learn about Intuition menus

AMICUS Disk 3

C programs:

Xref a C cross-reference gen., S-E
6bitcolor extra-half-bright chip gfx demo, S-E
Chop truncate (chop) files down to size, S-E
Cleanup removes strange characters from text files
CR2LF converts carriage returns to line feeds in Amiga files, S-E
Error adds compile errors to a C file, S
Hello window ex. from the RKM, S
Kermit generic Kermit implementation, flakey, no terminal mode, S-E
Scales sound demo plays scales, S-E
SkewB Rubik cube demo in hi-res colors, S-E
AmigaBasicProgs(dir)
Automata cellular automata simulation
CrazyEights card game

Graph function graphing programs

WitchingHour a game

ABasic programs:

Casino games of poker, blackjack, dice, and craps
Gomoku also known as 'othello'
Sabotage sort of an adventure game

Executable programs:

Disassem a 68000 disassembler, E-D
DpSlide shows a given set of IFF pictures, E-D
Arrange a text formatting program, E-D

Assembler programs:

Argoterm a terminal program with speech and Xmodem, S-E

AMICUS Disk 4 Files from the original Amiga Technical BBS

Note that some of these files are old, and refer to older versions of the operating system. These files came from the Sun system that served as Amiga technical support HQ for most of 1985. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

Complete and nearly up-to-date C source to 'image.ed', an early version of the Icon Editor. This is a little flaky, but compiles and runs.

An Intuition demo, in full C source, including files: demomenu.c, demomenu2.c, demoreq.c, getascil.c, idemo.c, idemo.guide, idemo.make, idemoall.h, nodos.c, and twrite.c

addmem.c add external memory to the system
bobtest.c example of BOB use
consoleIO.c console IO example
createport.c create and delete ports
creatstdl.c create standard I/O requests
creatask.c creating task examples
diskio.c example of track read and write
dotty.c source to the 'dotty window' demo
dualplay.c dual playfield example
flood.c flood fill example
freemap.c old version of 'freemap'
gettools.c tools for VSprites and BOBs
gfxmem.c graphic memory usage indicator
hello.c window example from RKM
inputdev.c adding an input handler to the input stream

joystik.c reading the joystick
keybd.c direct keyboard reading
layers.c layers examples
mousport.c test mouse port

ownlib.c example of making your own library with Lattice

paratest.c tests parallel port commands
seritest.c tests serial port commands
serisamp.c example of serial port use
printtr.c sample printer interface code
prtbase.h printer device definitions
regint.c region test program
setlace.c source to interlace on/off program
setparallel.c set the attributes of the parallel port
SetSerial.c set the attributes (parity, data bits) of the serial port

singplay.c single playfield example
speechtoy.c source to narrator and phonetics demo
timedely.c simple timer demo
timer.c exec support timer functions
timrstuf.c more exec support timer functions
WhichFont.c loads and displays all available system fonts

process.i and prbase.i assembler include files:

autorqstr.txt warnings of deadlocks with
 autorequesters
 consoleIO.txt copy of the RKM console I/O chapter
 diskfont.txt warning of disk font loading bug
 listfunc.txt list of #defines, macros, functions
 inputdev.txt preliminary copy of the input device
 chapter

License information on Workbench distribution license
 printer pre-release copy of the chapter on printer drivers,
 from RKM 1.1 v11fd.txt 'diff' of .ld file changes from
 version 1.0 to 1.1 v28v1.diff 'diff' of include file changes
 from version 28 to 1.0

AMICUS Disk 5. Files from the Amiga Link / Amiga Information Network

Note that some of these files are old, and refer to older
 versions of the operating system. These files are from
 Amiga Link. For a time, Commodore supported Amiga
 Link, aka AIN, for online developer technical support. It
 was only up and running for several weeks. These files
 do not carry a warranty, and are for educational purposes
 only. Of course, that's not to say they don't work.

A demo of intuition menus called 'menudemo', in C
 source

whereis.c find a file searching all subdirectories
 bobtest.c BOB programming example
 sweep.c sound synthesis example

Assembler files:

mydev.asm sample device driver
 mylib.asm sample library example
 mylib.i
 mydev.i
 asmsupp.i
 macros.i assembler include files:

Texts:

amigatricks tips on CLI commands
 extdisk external disk specification
 gameport game port spec
 parallel parallel port spec
 serial serial port spec
 v1.1.update list of new features in version 1.1
 v1.1h.txt 'diff' of include file changes from version
 1.0 to 1.1

Files for building your own printer drivers, including
 dospecial.c, eposdata.c, init.asm, printer.c, printer.link,
 printtag.asm, render.c, and wait.asm. This disk does
 contain a number of files describing the IFF specification.
 These are not the latest and greatest files, but remain
 here for historical purposes. They include text files and C
 source examples. The latest IFF spec is elsewhere in this
 library.

AMICUS Disk 6. IFF Pictures

This disk includes the DPSlide program, which can view
 a given series of IFF pictures, and the 'showpic' program,
 which can view each file at the click of an icon, and the
 'saveilbm' program, to turn any screen into an IFF picture.
 The pictures include a screen from ArticFox, a Degas
 dancer, the guys at Electronic Arts, a gorilla, horses, King
 Tut, a lighthouse, a screen from Marble Madness, the
 Bugs Bunny Martian, a still from an old movie, the Dire
 Straits moving company, a screen from Pinball
 Contruction Set, a TV newscaster, the PaintCan, a world
 map, a Porsche, a shuttle mission patch, a tyrannosaurus
 rex, a planet view, a VISA card, and a ten-speed.

AMICUS Disk 7. DigView HAM demo picture disk
 This disk has pictures from the DigView hold-and-modify
 video digitizer. It includes the ladies with pencils and
 lolypops, the young girl, the bulldozer, the horse and
 buggy, the Byte cover, the dictionary page, the robot and
 Robert. This includes a program to view each picture
 separately, and all together as separate, slidable screens.

AMICUS Disk 8

C programs:

Browse view text files on a disk, using menus
 S-E-D
 Crunch removes comments and white space
 from C files, S-E
 IconExec EXECUTE a series of commands from
 Workbench S-E
 PDScreen
 Dump dumps Rastport of highest screen to
 printer
 SetAlternate sets a second image for an icon, when
 clicked once S-E
 SetWindow makes windows for a CLI program to run
 under Workbench S-E
 SmallClock a small digital clock that sits in a window
 menu bar
 Scripmer the screen printer in the fourth Amazing
 Computing, S-E

Amiga Basic Programs:

(Note: Many of these programs are present on AMICUS
 Disk 1. Several of these were converted to Amiga Basic,
 and are included here.)

AddressBook a simple address book database
 Ball draws a ball
 Cload program to convert Compuserve hex files
 to binary, S-D
 Clue the game, intuition driven
 ColorArt art drawing program
 DeluxeDraw the drawing program in the 3rd issue of
 Amazing Computing, S-D

Eliza conversational computer psychologist
 the game, as known as 'go'
 RatMaze 3D ratmaze game
 ROR boggling graphics demo
 Shuttle draws 3D pictures of the space shuttle
 Spelling simple spelling program
 YoYo wierd zero-gravity yo-yo demo, tracks yo-
 yo to the mouse

Executable programs:

Modula-2 demo of a rotating cube
 sets a second icon image, displayed
 when the icon is clicked
 a slow but simple spelling checker, E-D
 the ARC file compression program,
 must-have for telecom, E-D
 graphics demo
 a program to rescue trashed disks, E-D
 a quick but nasty disk copy program:
 ignores errors, E-D
 lists hunks in an object file E-D
 saves any screen as an IFF picture
 E-D ??
 shareware screen dump program, E only
 version 2.0, term program, Xmodem
 E-D

Texts:

LatticeMain tips on fixing _main.c in Lattice
 GDiskDrive make your own 5 1/4 drive
 GuruMed explains the Guru numbers
 Lat3.03bugs bug list of Lattice C version 3.03
 MForgeRev user's view of the MicroForge hard drive
 PrintSpooler EXECUTE-based print spooling program

.BMAP files:

These are the necessary links between Amiga Basic and
 the system libraries. To take advantage of the Amiga's
 capabilities in Basic, you need these files. BMAPs are
 included for 'clist', 'console', 'diskfont', 'exec', 'icon',
 'intuition', 'layers', 'mathfp', 'mathleedoubas',
 'mathleedeesbas', 'mathtrans', 'potgo', 'timer' and
 'translator'.

AMICUS Disk 9

Amiga Basic Programs:

FlightSim simple flight simulator program
 HuePalette explains Hue, Saturation, and Intensity
 ex. of doing requesters from Amiga
 Basic
 ScrollDemo demonstrates scrolling capabilities
 Synthesizer sound program
 WorldMap draws a map of the world

Executable programs:

Boing! latest Boing! demo, with selectable
 speed, E
 Brush2C converts an IFF brush to C data
 instructions, initialization code, E
 Brush2Icon converts IFF brush to an icon, E
 Dazzle graphics demo, tracks to mouse, E
 DecIGEL assembler program for stopping 68010
 errors, S-E-D
 Klock menu-bar clock and date display, E
 life the game of life, E
 TimeSet intuition-based way to set the time and
 date,
 EMEmacs another Emacs, more oriented to word
 processing, S-E-D
 MyCLI a CLI shell, works without the
 Workbench, S-E-D

Texts:

FncdnKeys explains how to read function keys from
 Amiga Basic
 HackerSin explains how to win the game 'hacker'
 Ist68010 guide to installing a 68010 in your Amiga
 PrinterTip tips on sending escape sequences to
 your printer
 StartupTip tips on setting up your startup-
 sequence file
 XfrmReview list of programs that work with the
 Transformer

Printer Drivers:

Printer drivers for the Canon PJ-1080A, the C Itoh
 Prowriter, an improved Epson driver that eliminates
 streaking, the Epson LQ-800, the Gemini Star-10, the
 NEC 8025A, the Okidata ML-92, the Panasonic KX-P10xx
 family, and the Smith-Corona D300, with a document
 describing the installation process.

AMICUS Disk 10. Instrument sound demos

This is an icon-driven demo, circulated to many dealers.
 It includes the sounds of an acoustic guitar, an alarm, a
 banjo, a bass guitar, a boink, a callopie, a car horn,
 claves, water drip, electric guitar, a flute, a harp arpeggio, a
 kicddrum, a marimba, a organ minor chord, people
 talking, pigs, a pipe organ, a Rhodes piano, a saxophone,
 a star, a snare drum, a steel drum, bells, a vibrophone, a
 violin, a wailing guitar, a horse whinny, and a whistle.

AMICUS Disk 11

C programs

dirutil Intuition-based, CLI replacement file
 manager, S-E
 cprl shows and adjusts priority of CLI
 processes, S-E
 ps shows info about CLI processes, S-E
 vidtex displays Compuserve RLE pictures, S-E
 AmigaBasic programs
 pointeredit pointer and sprite editor program
 optimize optimization ex ample from AC article
 calendar large, animated calendar, diary and date
 book program
 amortize loan amortizations
 brushtobob converts small IFF brushes to
 AmigaBasic BOB OBJECTS
 grids draw and play waveforms
 hilbert draws Hilbert curves
 madlib mad lib story generator
 mailtalk talking mailing list program
 meadows3D 3D graphics program, from
 Amazing Computing article
 mousetrack mouse tracking example in hires mode
 slot slot machine game
 tictactoe the game
 switch pachinko-like game
 weird makes strange sounds
 Executable programs
 cp unix-like copy command, E
 cls screen clear, S-E
 diff unix-like stream editor uses 'diff' output
 to fix files
 pm chart recorder performances indicator
 Assembler programs
 cls screen clear and CLI arguments
 example
 Modula-2
 trails moving-worm graphics demo
 caseconvert converts Modula-2 keywords to
 uppercase
 Forth Breshehan circle algorithm example
 Analyze 12 templates for the spreadsheet
 Analyzel

There are four programs here that read Commodore 64
 picture files. They can translate Koala Pad, Doodle, Print
 Shop and News Room graphics to IFF format. Of course,
 getting the files from your C-64 to your Amiga is the hard
 part.

AMICUS Disk 12

Executable programs

blink 'alink' compatible linker, but faster, E-D
 clean spins the disk for use with disk cleaners,
 E-D
 eposset sends Epson settings to PAR: from menu,
 E-D
 showbig view hi-res pictures in low-res
 superbitmap, E-D
 speaktime tell the time, E-D
 undelete undeletes a file, E-D
 cnvapdihm converts Apple][low, medium and high
 res pictures to IFF, E-D
 menued menu editor produces C code for menus,
 E-D
 quick quick disk-to-disk nibble copier, E-D
 quickEA copies Electronic Arts disks, removes
 protection, E-D
 tred 1.3 demo of text editor from Microsmiths, E-D
 C programs
 spin3 rotating blocks graphics demo, S-E-D
 popcli start a new CLI at the press of a button,
 like Sidekick, S-E-D
 vsprite VSprite example code from Commodore, S-
 E-D
 AmigaBBS Amiga Basic bulletin board program, S-D

Assembler programs
 star10 makes star fields like Star Trek intro, S-E-D
 Pictures
 Mount Mandelbrot 3D view of Mandelbrot set
 Star Destroyer hi-res Star Wars starship
 Robot robot arm grabbing a cylinder
 Texts
 vendors list of Amiga vendors, names, addresses
 cardco fixes to early Cardco memory boards
 cinclude cross-reference to C include files, who
 includes what
 mindwalker clues to playing the game well
 slideshow make your own slideshows from the
 Kaleidoscope disk

AMICUS Disk 13

Amiga Basic programs

Routines from Carolyn Scheppler of CBM Tech Support, to read and display IFF pictures from Amiga Basic. With documentation. Also included is a program to do screen prints in Amiga Basic, and the newest BMAP files, with a corrected ConvertFD program. With example pictures, and the SaveILBM screen capture program.

Routines to load and play FutureSound and IFF sound files from Amiga Basic, by John Foust for Applied Visions. With documentation and C and assembler source for writing your own libraries, and interfacing C to assembler in libraries. With example sound.

Executable programs

gravity Sci Amer Jan 86 gravitation graphic
 simulation, S-E-D

Texts

MIDI make your own MIDI instrument interface,
 with documentation and a hi-res schematic
 picture.

AMICUS Disk 14

Several programs from Amazing Computing Issues:

Tools

Dan Kary's C structure index program, S-E-D

Amiga Basic programs

BMAP Reader by Tim Jones

IFFBrush2BOB by Mike Swinger

AutoRequester example

DOSHelper Windowed help system for CLI commands,
 S-E-D

PETrans translates PET ASCII files to ASCII files,
 S-E-D

C Squared Graphics program from Scientific American,
 Sept 86, S-E-D

crif adds or removes carriage returns from files,
 S-E-D

dpdecode decrypts Deluxe Paint, removes copy
 protection, E-D

queryWB asks Yes or No from the user, returns exit
 code, S-E

vc VisiCalc type spreadsheet, no mouse control,
 E-D

view views text files with window and slider
 gadget, E-D

Oing, Sproing, yaBoing, Zoing are sprite-based Boing! style
 demos, S-E-D

CLIClock, sClock, wClock are window border clocks, S-E-D

Texts

An article on long-persistence phosphor monitors, tips on
 making brushes of odd shapes in Deluxe Paint, and
 recommendations on icon interfaces from Commodore-Amiga.

Fred Fish Public Domain Software

Fred Fish Disk 1:

amigademo Graphical benchmark for comparing amigas.

amigaterm simple communications program with
 Xmodem

balls simulation of the "kinetic thingy" with balls
 on strings

colorful Shows off use of hold-and-modify mode.
 Dhrystone benchmark program.

doty Source to the "doty window" demo on the
 Workbench disk.

freedraw A small "paint" type program with lines,
 boxes, etc.

gad John Draper's Gadget tutorial program

gxmem Graphical memory usage display program

halfbrite demonstrates "Extra-Half-Brite" mode, if you
 have it

hello simple window demo

latftp accessing the Motorola Fast Floating Point I
 library from C

palette Sample program for designing color
 palettes.

trackdisk Demonstrates use of the trackdisk driver.

requesters John Draper's requester tutorial and
 example program.

speech Sample speech demo program. Stripped
 down "speechtoy".

speechtoy Another speech demo program.

Fred Fish Disk 2:

alib Object module librarian.

cc Unix-like frontend for Lattice C compiler.

dbug Macro based C debugging package.

Machine independent.

make Subset of Unix make command.

make2 Another make subset command.

microemacs Small version of emacs editor, with
 macros, no extensions

portar Portable file archiver.

xrf DECUS C cross reference utility.

Fred Fish Disk 3:

gothic Gothic font banner printer.

roff A "roff" type text formatter.

ff A very fast text formatter

cforth A highly portable forth implementation. Lots
 of goodies.

xlisp Xlisp 1.4, not working correctly.

Fred Fish Disk 4:

banner Prints horizontal banner

bgrep A Boyer-Moore grep-like utility

bison GNU Unix replacement 'yacc', not working.

bm Another Boyer-Moore grep-like utility

grep DECUS grep

kermit simple portable Kermit with no connect
 mode.

MyCLI Replacement CLI for the Amiga. Version 1.0

mandel A Mandelbrot set program, by Robert French
 and RJ Mical

Fred Fish Disk 5:

cons Console device demo program with
 supporting macro routines.

freemap Creates a visual diagram of free memory

input.dev sample input handler, traps key or mouse
 events

joystick Shows how to set up the gameport device
 as a joystick.

keyboard demonstrates direct communications with
 the keyboard.

layers Shows use of the layers library

mandelbrot IFF Mandelbrot program

mouse hooks up mouse to right joystick port

one.window console window demo

parallel Demonstrates access to the parallel port.

printer opening and using the printer, does a
 screen dump, not working

print.support Printer support routines, not working.

proctest sample process creation code, not working

region demos split drawing regions

samplefont sample font with info on creating your own

serial Demos the serial port

singlePlayfield Creates 320 x 200 playfield

speechtoy latest version of cute speech demo

simplified version of speechtoy, with IO
 requests

text.demo displays available fonts

timer demos timer.device use

trackdisk demos trakdisk driver

Fred Fish Disk 6:

compress like Unix compress, a file squeezer

dadac analog clock impersonator

microemacs upgraded version of microemacs from disk 2

mult removes multiple occurring lines in files

scales demos using sound and audio functions

setparallel Allows changing parallel port parameters

setserial Allows changing serial port parameters.

sortc quicksort based sort program, in C

stripc Strips comments and extra whitespace from
 C source

Fred Fish Disk 7:

This disk contains the executables of the game Hack,
 version 1.0.1.

Fred Fish Disk 8:

This disk contains the C source to Hack on disk 7.

Fred Fish Disk 9:

moire Draws moire patterns in black and white

MVP-FORTH Mountain View Press Forth, version
 1.00.03A. A shareware version of FORTH
 from Fantasia Systems.

proff a more powerful text formatting program

setface Program to toggle interface mode on and off.

skewb a rubic's cube type demo

sparks moving snake Graphics demo

Fred Fish Disk 10:

conquest An interstellar adventure simulation game

dehex convert a hex file to binary

filezap Patch program for any type of file.

fixobj Strip garbage off Xmodem transferred files.

iff Routines to read and write iff format files.

ld simple directory program

ls Minimal UNIX ls, with Unix-style wildcarding,
 in C

sq.usq file squeeze and unsqueeze

trek73 Star Trek game

yachtc Dice game.

Fred Fish Disk 11:

dpslide slide show program for displaying IFF
 Images with miscellaneous pictures

Fred Fish Disk 12:

amiga3d Shows a rotating 3 dimensional solid
 "Amiga sign".

ArgoTerm a terminal emulator program, written in
 assembler

arrow3d Shows a rotating 3 dimensional wire frame
 arrow.

ld4 directory listing program

IconExec

SetWindow two programs for launching programs from
 Workbench that presently only work under
 CLI.

SetAlternate Makes an icon show a second image when
 clicked once

StarTerm terminal emulator, with ASCII Xmodem,
 dialer, more.

Fred Fish Disk 13:

A Bundle of Basic programs, including:

Jpad toybox ezspeak mandelbrot

xmodem 3dsolids addbook algebra

ror amgseq1 amiga-copy band

bounce box brickout canvas

cardfi circle colorcircles Copy

cubes1 outpaste date dogstar

dragon draw dynamictriangle

Eliza ezterm fillbuster fractal

fscape gomoku dart haiku

ha9000 halley hauntedM hidden

join loz mandel menu

miniPaint mouse Orthello patch

pena pinwheel gbox random-circles

Readme rgb rgbtest Rord

sabotage salestalk shades shapes

shuffle sketchpad spaceart

speakspeech speecheasy spell

sphere spiral stripper superpad

supshr talk terminal termtest

tom topography triangle

wheels xenos xmostriper

(note: some programs are Abasic, most are Amigabasic, and
 some programs are presented in both languages)

Fred Fish Disk 14:

amiga3d update of #12, includes C source to a full
 hidden surface removal and 3D graphics

beep Source for a function that generates a beep
 sound

dex extracts text from within C source files

dimensions demonstrates N dimensional graphics

filezap update of disk 10, a file patch utility

gxmem update of disk 1, graphic memory usage
 Indicator

gi converts IFF brush files to Image struct, in C
 text.

pdterm simple ANSI VT100 terminal emulator,
 in 80 x 25 screen

shell simple Unix 'csh' style shell

termcap mostly Unix compatible 'termcap'
 implementation.

Fred Fish Disk 15:

Blobs graphics demo, like Unix 'worms'

Clock simple digital clock program for the title bar

Dazzle An eight-fold symmetry dazzler program.
 Really pretty!

Fish double buffered sequence cycle animation
 of a fish

Monopoly A really nice monopoly game written in
 Abasic.

Okdata Dump Okdata ML92 driver and WorkBench screen
 dump program.

Polydraw A drawing program written in Abasic.

Polyfractals A fractal program written in Abasic.

Fred Fish Disk 16:
Complete copy of the latest developer IFF disk

Fred Fish Disk 17:
The NewTek Digi-View video digitizer HAM demo disk

Fred Fish Disk 18:
AmigaDisplay dumb terminal program with bell, selectable fonts

Ash Pre-release C Shell-like shell program, history, loops, etc.

Browser wanders a file tree, displays files, all with the mouse

MC68010 docs on upgrading your Amiga to use a 68010

MultIdm rotate an N dimensional cube with a joystick

PigLatin SAY command that talks in Pig Latin

Schriper Screen image printer

Xlisp 1.6 source, docs, and executable for a Lisp interpreter.

Fred Fish Disk 19:
BlackJack text-oriented blackjack game

JayMinerSlides Slides by Jay Miner, Amiga graphics chip designer, showing flowchart of the Amiga Internals, in 640 x 400.

Keymap_Test test program to test the keymapping routines

LockMon Find unclosed file locks, for programs that don't clean up.

Fred Fish Disk 20:
AmigaToAtari converts Amiga object code to Atari format

DiskSalv program to recover files from a trashed AmigaDOS disk.

Hash example of the AmigaDOS disk hashing function

Hd Hex dump utility ala Computer Language magazine, April 86

MandelBros Mandelbrot contest winners

MultiTasking Tutorial and examples for Exec level multitasking

Pack strips whitespace from C source

PortHandler sample Port-Handler program that performs. Shows BCPL environment clues.

Random Random number generator in assembly, for C or assembler.

SetMouse2 sets mouse port to right or left port.

SpeechTerm terminal emulator with speech capabilities, Xmodem

TxEd Demo editor from Microsmiths Charlie Heath

Fred Fish Disk 21
This is a copy of Thomas Wilcox's Mandelbrot Set Explorer disk. Very good!

Fred Fish Disk 22
This disk contains two new "strains" of microemacs.

Lemac version 3.6 by Daniel Lawrence. For Unix V7, BSD 4.2, Amiga, MS-DOS, VMS. Uses Amiga function keys, status line, execute, startup files, more.

Pemac By Andy Poggio. New features include <ALT> keys as Meta keys, mouse support, higher priority, backup files, word wrap, function keys.

Fred Fish Disk 23
Disk of source for MicroEmacs, several versions for most popular operating systems on micros and mainframes. For people who want to port MicroEmacs to their favorite machine.

Fred Fish Disk 24:
Conques installer adventure simulation game update to shell on Disk 14, with built in commands, named variables, substitution.

Csh A pre-release version of the single pass Modula-2 compiler originally developed for Macintosh at ETHZ. This code was transmitted to the AMIGA and is executed on the AMIGA using a special loader. Binary only.

Fred Fish Disk 25
Graphic Hack A graphic version of the game on disks 7 and 8

Fred Fish Disk 25
This is the graphics-oriented Hack game by John Toebes. Only the executable is present.

Fred Fish Disk 26
UnHunk Processes the Amiga "hunk" loadfiles. Collect code, data, and bss hunks together, allows individual specification of code, data, and bss origins, and generates binary file with format reminiscent of Unix "a.out" format. The output file can be easily processed by a separate program to produce Motorola "S-records" suitable for downloading to

PROM programmer. By Eric Black.

C-kermit Port of the Kermit file transfer program and server.

Ps Display and set process priorities

Archx Yet another program for bundling up text files and mailing or posting them as a single file unit.

Fred Fish Disk 27
ABdemos Amiga Basic demos from Carolyn Schepner.

NewConvertFD creates .bmaps from fd files.

BitPlanes finds addresses of and writes to bitplanes of the screen's bitmap.

AboutBmaps A tutorial on creation and use of bmaps.

LoadILBM loads and displays IFF ILBM pics.

LoadACBM loads and displays ACBM pics.

ScreenPrint creates a demo screen and dumps it to a graphic printer.

Disassem Simple 68000 disassembler. Reads standard Amiga object files and disassembles the code sections. Data sections are dumped in hex. The actual disassembler routines are set up to be callable from a user program so instructions in memory can be disassembled dynamically. By Bill Rogers.

DvorakKeymap Example of a keymap structure for the Dvorak keyboard layout. Untested but included because assembly examples are few and far between. By Robert Burns of C-A.

Hypocycloids Spirograph, from Feb. 84 Byte.

LinesDemo Example of proportional gadgets to scroll a SuperBitMap.

MemExpansion Schematics and directions for building your own homebrew 1 Mb memory expansion, by Michael Fellingner.

SafeMalloc Program to debug 'malloc()' calls

ScienceDemos Convert Julian to solar and sidereal time, stellar positions and radial velocity epoch calculations and Galilean satellite plotter. By David Eagle.

Fred Fish Disk 28
ABasic games by David Addison: Backgammon, Cribbage, Milestone, and Othello

C++ DECUS 'cpp' C preprocessor, and a modified 'cc' that knows about the 'cpp', for Manx C.

Shar Unix-compatible shell archiver, for packing files for travel.

SuperBitMap Example of using a ScrollLayer, syncing SuperBitMaps for printing, and creating dummy RastPorts.

Fred Fish Disk 29
AegisDraw Demo Demo program without save and no docs.

Animator Demo Player for Aegis Animator files

Cc Unix-like front-end for Manx C.

Enough Tests for existence of system resources, files, devices.

Rubik Animated Rubik's cube program

StringLib Public domain Unix string library functions.

Vt100 VT-100 terminal emulator with Kermit and Xmodem protocols

Fred Fish Disk 30
Several shareware programs. The authors request a donation if you find their program useful, so they can write more software.

BBS an Amiga Basic BBS by Ewan Grantham

FineArt Amiga art

FontEditor edit fonts, by Tim Robinson

MenuEditor Create menus, save them as C source, by David Pehrson

StarTerm3.0 Very nice telecommunications by Jim Nangano

Fred Fish Disk 31
Life Life game, uses blitter to do 19.8 generations a second.

Mandelbrot Version 3.0 of Robert French's program.

MxExample Mutual exclusion gadget example.

RamSpeed Measure relative RAM speed, chip and fast.

Set Replacement for the Manx "set" command for environment variables, with improvements.

Tree Draws a recursive tree, green leafy type, not files.

TxEd Crippled demo version of Microsmith's text editor, TxEd.

VDRAW Full-featured drawing program by Stephen Vermeulen.

XIcon Invokes CLI scripts from icon

TIcon Displays text files from an icon.

Fred Fish Disk 32
Address Extended address book written in AmigaBasic.

Calendar Calendar/diary program written in AmigaBasic.

DosPlus1 First volume of CLI oriented tools for developers.

DosPlus2 Second volume of CLI oriented tools for developers.

Executables only.

MacView Views MacPaint pictures in Amiga low or high res, no sample pictures, by Scott Evernden.

Puzzle Simulation of puzzle with moving square tiles.

ShowHAM View HAM pictures from CLI.

Solitaire ABASIC games of Canfield and Klondike, from David Addison.

Spin3 Graphics demo of spinning cubes, double-buffered example.

Sword Sword of Fallen Angel text adventure game written in Amiga Basic.

Trails Leaves a trail behind mouse, in Modula-2

Fred Fish Disk 33
3dstars 3d version of the "stars" program below.

Bigmap Low-level graphics example scrolls bitmap with ScrollVPort.

Dbuf.gels Double-buffered animation example for BOBs and VSprites.

DiskMapper Displays sector allocation of floppy disks.

MemView View memory in real time, move with joystick.

Oing Bouncing balls demo

Sproing Oing, with sound effects.

ScreenDump Dumps highest screen or window to the printer.

Sdb Simple database program from a DECUS tape.

Stars Star field demo, like Star Trek.

TermPlus Terminal program with capture, library, function keys, Xmodem, CIS-B protocols.

Vt100 Version 2.0 of Dave Wecker's VT-100 emulator, with scripts and function keys.

Fred Fish Disk 34
Alint Support files for Gimpel's 'lint' syntax checker

Blink PD 'alink' compatible linker, faster, better.

Browser Updated to FF 18 'browser', in Manx, with scroll bars, bug fixes.

Btree b-tree data structure examples

Btree2 Another version of 'btree'

Calendar Appointment calendar with alarm.

Less File viewer, searching, position by percent, line number.

NewFonts Set of 28 new Amiga fonts from Bill Fischer

Pr Background print utility, style options, wildcards.

Requester Deluxe Paint-type file requester, with sample.

Fred Fish Disk 35
ASendPacket C example of making asynchronous I/O calls to a DOS handler, written by C-A.

ConsoleWindow C example of getting the Intuition pointer a CON: or RAW: window, for 1.2, by C-A.

DirUtil Walk the directory tree, do CLI operations from menus

DirUtil2 Another variant of DirUtil.

FileRequester Lattice C file requester module, with demo driver, from Charlie Heath.

MacView Views MacPaint pictures in Amiga low or high res, with sample pictures, by Scott Evernden.

Plop Simple IFF reader program

PopCLI Sidekick-style program invokes a new CLI, with automatic screen blanking.

QuickCopy Devenport disk copiers duplicate copy-protected disks.

ScrollPf Dual playfield example, from C-A, shows 400 x 300 x 2 bit plane playfield on a 320 x 200 x 2 plane deep playfield.

SendPacket General purpose subroutine to send AmigaDos packets.

SpriteMaker Sprite editor, can save work as C data structure. Shareware by Ray Larson.

Tracker Converts any disk into files, for electronic transmission. Preserves entire file structure. Shareware by Brad Wilson.

TriCops 3-D space invasion game, formerly commercial, now public domain. From Geodesic Publications.

Tsize Print total size of all files in subdirectories.

Unldef C preprocessor to remove given #ldef'd sections of a file, leaving the rest alone. By Dave Yost.

Vttest VT-100 emulation test program. Requires a Unix system.

Fred Fish Disk 36

Acp Unix-like 'cp' copy program
 Clock Updated version of clock on disk 15.
 Csh Manx 'csh'-like CLI, history, variables, etc.
 DietAid Diet planning aid organizes recipes, calories
 Echo Improved 'echo' command with color, cursor addressing
 FixHunk Fixes programs to let them run in external memory.
 Fm Maps the sectors a file uses on the disk.
 KickBench Docs, program to make a single disk that works like a Kickstart and Workbench.
 Lex Computes Fog, Flesch, and Kincaid readability of text files.
 TunnelVision David Addison ABasic 3D maze perspective game.
 Vc Visicalc-like spreadsheet calculator program.
 Vt100 Version 2.2 of Dave Wecker's telecom program
 YaBoing Oling style game program shows sprite collision detects

Fred Fish Disk 37

This disk is a port of Timothy Budd's Little Smalltalk system, done by Bill Kinnersley at Washington State University.

Fred Fish Disk 38

CSquared Sep 86 Sci American, Circle Squared algorithm
 FixObj Strips garbage off Xmodem transfered object files
 Handler AmigaDOS handler (device) example from C-A
 Hp-10c Mimics a HP-10C calculator, written in Modula-2
 IFFencode Saves the screen as an IFF file
 IftDump Dumps info about an IFF file
 Jsh BDS C-like CLI shell
 NewStat STATUS-like program, shows priority, processes
 Reversi Game of Reversi, version 6.1
 UUdecode Translate binary files to text, Unix-like programs
 Vdraw Drawing program, version 1.14
 VoiceFilter DX MIDI synthesizer voice filter program
 Window Example of creating a DOS window on a custom screen

Fred Fish Disk 39

AnsiEcho 'echo', 'touch', 'list', 'cis' written in assembler.
 Display Displays HAM images from a ray-tracing program, with example pictures.
 Driver Example device driver source, acts like RAM: disk
 Xlisp XLisp 1.7, executable only

Fred Fish Disk 40

Ahost Terminal emulator with Xmodem, Kermit and CIS B protocols, function keys, scripts, RLE graphics and conference mode.
 AmigaMonitor Dynamically displays the machine state, such as open files, active tasks, resources, device states, interrupts, libraries, ports, etc.
 Arc Popular file compression system, the standard for transitting files
 AreaCode Program that decodes area codes into state and locality.
 Blink 'alink' replacement linker, version 6.5
 Cosmo An 'asteriods' clone.
 Dg210 Data General D-210 Terminal emulator
 DirUtil Windowed DOS interface program, version 1.4
 DOSHelper Windowed AmigaDOS CLI help program
 PagePrint Prints text files with headers, page breaks, line numbers
 PopCLI Starts a new CLI with a single keystroke, from any program, With a screen-saver feature. Version 2, with source.
 SpriteEd Sprite Editor edits two sprites at a time
 X-Spell Spelling checker allows edits to files

(Fred Fish Disk#30 is free when ordered with at least three other disks from the collection.)

In Conclusion

To the best of our knowledge, the materials in this library are freely distributable. This means they were either publicly posted and placed in the Public Domain by their Author, or they have restrictions published in their files to which we have adhered. If you become aware of any violation of the author's wishes, please contact us by mail.

•AC•

Need AMIGA Software?

Try The Public Domain.....



Amazing Computing™ has vowed, from our begining, to amass the largest selection of Public domain software in the Amiga Community, and with the help of John Foust and Fred Fish, we see a great selection of software for both beginners and advanced users.

These Public Domain software pieces are presented by a world of authors who discovered something fun or interesting on the Amiga and then placed their discoveries in the Public Domain for all to enjoy. You are encouraged to copy and share these disks and programs with your friends, customers and fellow user group members!

The disk are very affordable!

Amazing Computing™ subscribers.....\$6.00 per disk.
 Non subscribers.....\$7.00 per disk

This is extremely reasonable for disks with almost 800K of information and programs. If you agree, please send check or money order to:

PiM Publications Inc.
 P.O. Box 869
 Fall River, MA 02722

All Checks must be in US funds drawn on a US Bank

Please allow 4 to 6 weeks for delivery

Amazing Computing™: Your resource to the Commodore Amiga

Amazing ComputingTM

If you are reading our *Amazing Directory* without seeing *Amazing Computing*TM, look what you are missing in the rest of Volume 1 Number 9:

Dos 2 Dos reviewed by Richard Knepper Transfer files from PC/MS-DOS and AmigaBasic
MaxiPlan reviewed by Richard Knepper The Amiga version of Lotus 1-2-3
Gizmoz by reviewed by Peter Wayner A collection of Amiga extras!
The Loan Information Program by Brian Catley A basic program to "review" your financial options
Starting Your Own Amiga Related Business by William Simpson The possible ways to establish your business.
Keep Track of Your Business Usage for Taxes by James Kummer A program to justify your Amiga to the IRS
The Absoft Amiga Fortran Compiler reviewed by Richard A. Reale Use your valuable Fortran programs.
Using Fonts from AmigaBasic, Part Two by Tim Jones The Amiga Basic program outlined last issue
68000 Macros on the Amiga by Gerald Hull Advance your program's ability.
TDI Modla-2 Amiga Compiler by Steve Faiwiszewski Looking at an alternative to C and Forth.

Or, what you have missed in our previous issues!

Volume 1 Number 1 Premiere February 1986

Super Spheres By Kelly Kauffman An ABasic Graphics program
Date Virus By John Foust There is a disease that may attack your Amiga
EZ-Term by Kelly Kauffman An ABasic Terminal program
Miga Mania by Perry Kivolowitz Programming fixes and mouse care
Inside CLI by George Musser a guided insight into the AmigaDosTM
CLI Summary by George Musser Jr. A removable list of CLI commands
AmigaForum by Bela Lubkin A quick trip through Compuserve's Amiga SIG
Commodore Amiga Development Program by Don Hicks What to ask and where to go to be a developer
Amiga Products A listing of present and expected products.

Volume 1 Number 2 March 1986

Electronic Arts Comes Through A look at the new software from EA
Inside CLI: part two by George Musser George continues his investigation of CLI and ED
A Summary of ED Commands
Live! by Rich Milner A review of the Beta version of the Live! frame grabber
Online and the CTS Fabite 2424 ADH Modem by John Foust
Amiga Products
Superterm V 1.0 By Kelly Kauffman A terminal program written in Amiga Basic
A Workbench "More" Program by Rick Wirch
Amiga BBS numbers

Volume 1 Number 3 April 1986

Analyze! a review by Ernest Viveiros
Reviews of Racter, Barataccas and Mindshadow
Forth! The first of our on going tutorial
Deluxe Draw! by Rich Wirch An Amiga Basic program for the artist in us all.
Amiga Basic, A beginners tutorial The start of our tutorial of the most active Amiga language.
Inside CLI: part 3 by George Musser George gives us PIPE

Volume 1 Number 4 May 1986

SkyFox and ArticFox Reviewed
Build your own 5 1/4 Drive Connector By Ernest Viveiros
Amiga Basic Tips by Rich Wirch
Scripper Part One by Perry Kivolowitz A C program to print your Amiga screen
Microsoft CD ROM Conference by Jim O'Keane
Amiga BBS Numbers

Volume 1 Number 5 1986

The HSI to RGB Conversion Tool by Steve Pietrowicz a basic program for color manipulation
AmigaNotes by Rick Rae The first of the Amiga music columns
Sidecar A First Look by John Foust A first "under the hood" look at the IBM compatible hardware
John Foust Talks with R. J. Mical at COMDEXTM
How does Sidecar affect the Transformer an interview with Douglas Wyman of Simile
The Commodore Layoffs by John Foust John looks at the "cuts" at Commodore
Scripper Part Two by Perry Kivolowitz
Marauder reviewed by Rick Wirch
Building Tools by Daniel Kary

Volume 1 Number 6 1986

Temple of Apschal Trilogy reviewed by Stephen Pietrowicz
The Halley Project: A Mission in our Solar System reviewed by Stephen Pietrowicz
Flow: reviewed by Erv Bobo
Textcraft Plus a First Look by Joe Lowery
How to start your own Amiga User Group by William Simpson
Amiga User Groups
Mailing List by Kelly Kauffman a basic mail list program
Pointer Image Editor by Stephen Pietrowicz
Scripper: part three by Perry Kivolowitz
Fun With the Amiga Disk Controller by Thom Sterling
Optimize Your AmigaBasic Programs for Speed by Stephen Pietrowicz

Volume 1 Number 7 1986

Aegle Draw: CAD comes to the Amiga by Kelly Adams
Try 3D by Jim Meadows an introduction to 3D graphics
Aegle Images/ Animator: a review by Erv Bobo
Deluxe Video Construction Set reviewed by Joe Lowery
Window requesters in Amiga Basic by Steve Michel
ROT by Colin French a 3D graphics editor
"I C What I Think" Ron Peterson with a few C graphic programs
Your Menu Sir! by Bryan D. Catley programming menus in Amiga Basic
IFF Brush to AmigaBasic 'BOB' editor by Michael Swinger Convert IFF Brush Files for use with Amiga Basic
Linking C Programs with Assembler Routines on the Amiga by Gerald Hull

Volume 1 Number 8 1986

The University Amiga By Geoff Gamble Amiga's inroads at Washington State University
MicroEd a look at a one man army for the Amiga
MicroEd, The Lewis and Clark Expedition reviewed by Robert Frizelle
Scribble Version 2.0 a review
Computers in the Classroom by Robert Frizelle
Two for Study by Robert Frizelle a review of Discovery and The Talking Coloring Book
True Basic reviewed by Brad Grier
Using your printer with the Amiga
Marble Madness reviewed by Stephen Pietrowicz
Using Fonts from AmigaBasic by Tim Jones
Screen SeVer by Perry Kivolowitz A monitor protection program in C
Lattice MAKE Utility reviewed by Scott P. Evernden
A Tale of Three EMACS by Steve Polling
.bmap File Reader in Amiga Basic by Tim Jones A look into the .bmap files

Your Resource to the Commodore Amiga™

Plus, don't forget our regular columns:

The Amicus Network (a "Newsletter" of the Amiga Computer Users)
AmigaNotes (a music column)
ROOMERS (an insider's look at the Amiga Development Community)
Forth!
The Amazing C Tutorial

Amazing Computing has been offering the Amiga community the best in technical knowledge and reviews for the Commodore-Amiga™ since our first issue in February 1986.

We were the first magazine to document CLI
We were the first to show Sidecar™ from COMDEX™ in full detail.
We were the first to document a 5 1/4 drive connector
We were the first magazine to offer serious programming examples and help.
We were the first magazine to offer Public Domain Software at reasonable prices.
We were the first magazine with the user in mind!

However, Amazing Computing™ will not rest on past achievements. The Commodore-Amiga™ has more surprises for you and we are ready to cover them. We even have a few tricks that will "Amaze" you.

To subscribe to Amazing Computing™, please fill out the form below and send to: PiM Publications Inc., P.O.Box 869, Fall River, MA 02722. Back issues are still available at \$4.00 each

Yes! Amaze Me!

Please start my subscription to Amazing Computing™ with the next available issue. I have enclosed \$24.00 for 12 issues in the U.S. (\$30.00 Canada and Mexico, \$35.00 overseas) All funds must be in U.S. Currency drawn on a U.S. Bank.

Please circle your selection

Subscription PDS (as noted) Back Issues

Public Domain Software:

\$6.00 each for subscribers (yes, even new ones!)

\$7.00 each for non subscribers.

Name _____

Street _____

City _____ St. _____ ZIP _____

Amount enclosed _____

AMICUS:	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13																											
Fred Fish:	FF1	FF2	FF3	FF4	FF5	FF6	FF7	FF8	FF9	FF10	FF11	FF12	FF13	FF14	FF15	FF16	FF17	FF18	FF19	FF20	FF21	FF22	FF23	FF24	FF25	FF26	FF27	FF28	FF29	FF30	FF31	FF32	FF33	FF34	FF35	FF36	FF37	FF38	FF39	FF40

BACK ISSUES: \$4.00 each

VOL.1#1 VOL.1#2 VOL.1#3 VOL.1#4 VOL.1#5 VOL.1#6 VOL.1#7

Mass. Residents, please add 5% sales tax on PDS orders

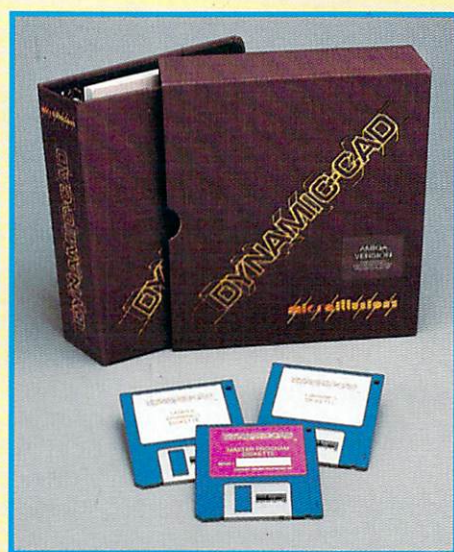
engineers, architects and designers are deserting their drafting tables for the precision and ease of

DYNAMIC-CAD

THE PROVEN CAD SYSTEM FOR THE AMIGA®

The highly advanced and powerful DynamicCad Drafting System by Microillusions has recently emerged from years of successful applications as a professional CAD system in the aerospace and piping industries. Combined with the Commodore Amiga, the most dynamic and versatile microcomputer on the market today, DynamicCad is revolutionizing the work methods of countless engineers and architects. DynamicCad's time and money-saving applications for these highpowered professionals is truly astounding. Here is an advanced, 2-D drafting system with isometric capabilities that can be combined with many models of plotters, printers, and digitizers.

The DynamicCad software was developed with three overriding principals in mind. First, it had to be 'easy to learn,' which resulted in DynamicCad's simple commands and abundant help tools. Next, it had to be 'easy to use,' which was assured by DynamicCad's powerful tools,



simple commands, and mouse menu functions, which combine to make the revising and capturing of drawings quick and efficient. Finally, DynamicCad had to provide great flexibility. This has been

achieved by its efficiency in producing every type of mechanical and architectural drawing, including printed circuit boards, integrated circuit designs, pert charts, piping diagrams, and electrical diagrams with their associated net lists. With the addition of extra memory DynamicCad will provide you with what may be the fastest PC type CAD system available.

Microillusions has an excellent upgrade policy and any changes to the system software will also be made available to existing DynamicCad users. Upcoming features for DynamicCad include a hierarchical data base which will allow for increased flexibility for underlying relevant information on library parts. New libraries will be added, and additional plotter and printer drivers. Currently Microillusions is writing a template for use on most digitizers. As the capabilities of DynamicCad expand our newsletter will help you keep pace with the technology and grow along with it.

DYNAMIC-CAD IS A PRODUCT OF



INQUIRIES TEL. (818) 360-3715

P. O. BOX 3475, GRANADA HILLS, CA 91344

DYNAMIC-CAD SYSTEM FEATURES:

DC Automatically configures itself to support additional memory
Supports most printers and plotters
Supports hard disk systems
DC is not copy protected
DC supplies online help
Screen resolutions of 640 x 400 and 640 x 200 modes
Both keyboard and mouse functions

- Extensive symbol library
- Alphanumerics: left, right, center, horizontal, vertical, varied angle
- Multiple line possibilities with varied arcs and degrees
- Horizontal & verticle doglegs
- Automatic line dimensioning in U.S. standard, metric or neither
- Gives X Y coordinates
- Create own pseudo symbols

- Arcs and circles
- Editing commands to move, delete and search
- Enter, rotate, change size or delete pseudos
- Group functions to manipulate, delete, step and repeat, move, etc.
- Fill and cross hatch capabilities
- Zoom and move elements around, resizing and repositioning

- Creates automatic schematics
- Creates net lists from electronic drawing or schematic
- Parameter settings include window size, percent of viewing area, alphanumeric ratios, 8,192 level selections, adjustable grid sizes, third line showing, grid set and overlay, line snap, alpha size, off screen display
- Can capture pictures in IFF format

AMIGA IS A TRADEMARK OF COMMODORE BUSINESS MACHINES, LTD.

The Absoft Amiga FORTRAN Compiler

"Don't throw away those old punched cards
with your classic FORTRAN programs. "

Reviewed by
Richard A. Reale

The Absoft Corporation has ported their complete implementation of the 1977 ANSI version of the FORTRAN language to the Amiga. Termed AC/FORTRAN, the compiler is a direct descendent of Absoft's other microcomputer implementations, such as the Macintosh, Atari ST, Hewlett Packard. It is completely source compatible with their mainframe FORTRAN compilers.

The compiler supports a number of language extensions which will probably be standard in the next revision of FORTRAN, as well as productivity tools including a full-screen interactive debugger, a linker, a library manager and a C interface. There is also a version of the AC/FORTRAN environment which supports the Motorola 68881 floating point chip in the Turbo Amiga.

In keeping with tradition, the Absoft compiler has pre-connected FORTRAN logical unit number (LUN) 5 for program data to the simulated card reader, which is simply a filename supplied on the command line at run time.

In the same manner, LUN 6 is pre-connected to the simulated line printer which is a dynamically generated output file and is automatically spooled to your printer. So, for many scientists and engineers, the transition from mainframe to microcomputer FORTRAN may be no more complex than transferring their source code and input data to text files on the Amiga.

This review of AC/FORTRAN was done on an Amiga 1000 running under Workbench 1.1. This Amiga has a retro-fit 68010 CPU instead of the standard 68000, 512K of RAM and a single internal 3 1/2 inch disk drive. Occasionally, some test programs were run on a standard Amiga with the 68000 CPU, but running under a beta version of the soon to be released Workbench 1.2. The test results showed only minor differences between these two environments.

In this report, I concentrated on an evaluation of the compilers adherence to the ANSI 77 standards for FORTRAN and on the veracity of the instructions for the compiler's use.

A future article might more fully analyze the performance of specific programs with regard to the operation of the run-time environment, the library manager and the FORTRAN interface with C, assembly language and Amiga ROM routines.

Absoft Reference Manual

The AC/FORTRAN reference manual has no doubt benefitted from its previous personal computer incarnations. It is

professionally printed, sub-divided and reads quite easily. The manual is composed of three main sections occupying about 300 pages. A seemingly generic section describes the mechanics of the compiler's invocation and the actions produced by an array of compile-time options.

Some of the most useful options permit the source code to be compiled under FORTRAN 66 standards, or to set variable size defaults to INTEGER*2 and LOGICAL*2, or to adjust the working heap size in multiples of 1024 bytes. Another chapter describes the operation of their symbolic debugger.

Six additional chapters, which complete this generic section, form the body of a typical FORTRAN 77 reference volume. Absoft is quick to point out that this material is for reference only. It is not meant to be a FORTRAN tutorial for the novice programmer.

A second section deals with the specific implementation on the Amiga. This material is also praiseworthy both for its clarity and content. It includes advice on the setup of a development Workbench disk suggesting the location, content and names of working directories which the compiler expects to access. Additionally, there is a chapter devoted to the FORTRAN 77 interface to the Amiga ROM kernel routines.

For those of us who lacked the ambition or patience to buy the ROM kernel manuals, this chapter is a real bonus. It summarizes, one by one, almost all the ROM routines, including those in the DOS, Exec, graphics and Intuition libraries.

The final section describes the operation of Absoft's FORTRAN linker which performs a permanent linking of incomplete object modules into an executable program, and the use of Absoft's FORTRAN library manager, which assimilates collections of FORTRAN modules under the user's direction.

Compiling FORTRAN programs

At first, the operation of the compiler may be a little confusing to a mainframe programmer. The compiler can generate an executable task without the subsequent use of a linker. This will automatically occur when all references and procedures named in the main program module are located by the compiler in its routine search path. Any additional system information normally provided by the linker on a mainframe is communicated dynamically by the Absoft run-time environment on the Amiga.

The compiler is said to be 'disk-based,' meaning that even large program compilation can be virtually free of excessive memory

consumption. Even on a single disk system, it is not overly inconvenient to keep all the source code on one disk and the compiler and its associated files on another Workbench disk. In this configuration, several disk switches are necessary for successful compilation. Generally, the "not enough memory" Guru message will be avoided.

On the other hand, I like to live dangerously and copy the source code to RAM: where the compilation proceeds 4-5 times faster and with no disk swaps. The compiler's output is position-independent and reentrant. Thus, a FORTRAN compiled module may be loaded anywhere in the Amiga's memory and may be shared by several programs simultaneously.

The actual invocation of the compiler, which is called 'F77' on the Absoft distribution disk, is made on a command line in any current CLI window. Compile time options are expressed with Unix-style command lines, and precede the source file name. To compile the file TEST.FOR for example, the following could be entered:

```
1> F77 -L -S -U -Z30 TEST.FOR
```

The L option produces a standard listing file with error diagnostics to be used during program development. The S option produces a symbol table file which is required for the operation of the debugger tool to be discussed below. The U option redirects input and output to the terminal if an asterisk (*) was used to specify these LUNs in the source code.

Finally, the Z option specifies the number of kilobytes of heap space allocated for this compilation. If the Z option is omitted, the default is 20 K. This may be insufficient for the compilation of large programs.

The compiler will produce a display on the terminal which reports on the status of the compilation and which is partially dictated by the options selected on the command line. The display shown below was produced by the example command line above. The example source code TEST.FOR had an intentional syntax error to demonstrate the effect on this display.

Absoft FORTRAN 77 Compiler Version 2.2

```
1: Symbol table complete - 1 error detected
```

Memory usage:

```
Labels    600 bytes
Symbols 3480 bytes
Total      51454 bytes
Excess     102374 bytes
Source  1117 lines
```

```
2: Bypassing
```

```
3: List file complete
```

```
4: Bypassing
```

When the error in the source file was corrected and the file was compiled, the following display was produced:

Absoft FORTRAN 77 Compiler Version 2.2

```
1: Symbol table complete
```

Memory usage:

```
Labels    600 bytes
Symbols 3480 bytes
Total      51454 bytes
Excess     63390 bytes
Source  1117 lines
```

```
2: Object file complete
```

```
3: List file complete
```

```
4: DEBUG symbol file complete
```

```
5: Program file complete: 23616 bytes
```

```
Elapsed time: 0:52 = 1288 lines/minute
```

Testing the compiler

In all, I selected and compiled over 50 individual FORTRAN programs. About three-quarters of these were specifically written to test whether particular features of the ANSI specification for FORTRAN 77 were supported in Absoft's product. Nearly 2000 individual features were tested and all were found to be supported.

I heard a rumor that suggested the compiler was having problems with VIRTUAL ARRAYS or with normal ARRAYS in deeply nested DO loops. Virtual arrays are another way to save on program memory usage since these arrays are stored on disk rather than in memory. I specifically tested for problems associated with manipulations of FORTRAN arrays, but could find none.

I do know of a situation in which even a properly compiled program will fail. The Absoft reference manual warns "If the stack is not large enough, the program may execute erratically, possibly causing a system failure (a crash)".

By "erratically," they mean that sometimes an executable program will run and sometimes it will not run, even if all other things appear equal. If the stack is too small, it is also possible for different Guru meditation numbers to be produced by different invocations of the same program!

Absoft suggests that the initial STACK size of 4 Kbytes allocated by default for a particular CLI session should be increased to the sum of the storage requirements for each program unit shown in the listing file. While this may seem quite straightforward, in practice I often had to set the stack STACK to be twice larger than this estimate, for consistent success.

It is apparent that the current Absoft product is aimed at the scientific and engineering workplace. An increasing number of scientists, engineers and other FORTRAN programmers are utilizing personal computers in place of minicomputers for their daily programming environments.

With this in mind, I compared the efficiency of the Amiga as a personal FORTRAN workstation with that of a typical multiuser-multitasking minicomputer. To do this all of the test programs were compiled on both an Amiga and on a DEC PDP-11/23 machine. Every effort was made to minimize the total time to produce an executable image.

Thus, on the Amiga all of the FORTRAN source code was placed into RAM: and no other tasks or processes were running during the compilation. Likewise, on the PDP-11/23, all of the executable files were built while only one user was on the system, a rather unnatural state for a typical minicomputer installation.

Nevertheless, the Absoft FORTRAN environment on the Amiga was a clear winner with an average waiting time of only 15 seconds compared to 95 seconds on the minicomputer. The somewhat small average size (8.5K) of the resulting executable test files on the Amiga (range from 3K to 25K) did tend to emphasize the overhead of the minicomputer. On the other hand, the differential would clearly have been even larger if the usual number of people (2 to 5) were all working on the minicomputer.

I was also interested in comparing the speed and accuracy of a mathematically intensive FORTRAN program on the Amiga with its homolog on another computer.

However, it did not seem justified to restrict this comparison to the minicomputer since most have dedicated floating point processors and their speed advantage is assured.

For the present I choose to use two benchmarks which were recently tested on a Macintosh (see "Fortran's New Life on the Mac", E. Floack and M. Flock, MacWorld, June 1986) after being compiled with Microsoft's FORTRAN version 2.1 developed by Absoft. All calculations were done using double-precision floating point variables which on the Amiga assures that values are treated as 64 bit entities and follow the IEEE proposed standard (task P745).

The first benchmark calculated the product of $B \cdot B$, where:

$$B = \tan(\arctan(\exp(\log(\sqrt{A \cdot A}))))/A - 1$$

and A is incremented from 1 to 2500. The total execution time on the Amiga (103.8 sec) was almost two times that on the minicomputer (50.4 sec) and fully 23 percent greater than the reported value for the Macintosh (84.3 sec).

In comparison, the root mean square error of the calculation on the Amiga ($2.3E-15$) was almost twice as great as that on the minicomputer ($1.2E-15$), but only about one-thousandth of the value reported for the Macintosh ($1.1E-10$).

The second benchmark calculated the sin, cos, tan, arctan, and log of the numbers from 1 to 10,000.

Again, the Amiga's 308 seconds appeared significantly slower than both the minicomputer (231 sec) and the published time of 108.6 sec for the Macintosh.

However, I have verified the times for the Macintosh were based on single-precision arithmetic. If single-precision arithmetic is used on the Amiga, the elapsed times for each benchmark would be 23.1 sec and 94.1 sec, respectively.

Linking of compiled object modules

Absoft has provided two mechanisms for linking a number of compiled program modules into a single executable image file. The traditional approach uses their linker. It resolves all

references between the main program module and the list of additional modules to produce a static and permanently linked executable image.

The second approach is a powerful overlay procedure. The main program unit and all additional modules are compiled individually, but are not statically linked into a single permanent image.

Rather, the Absoft FORTRAN run-time system performs a dynamic linking of program modules as each is needed. This latter approach is particularly attractive for large programs where memory usage of a single task may be prohibitive and is actually less complicated than static linking to achieve.

This simplification occurs because the Absoft run-time environment can provide for the default automatic searching and loading of unlinked program units. To do this, Absoft has defined a default search path for the dynamic linking. It looks first in the current directory, then to the library directory of the current disk, and finally to the library directory of the currently assigned system disk.

The memory space allocated for a given program module's execution and data storage is recovered when the RETURN statement of that module is executed.

Thus, by careful program design, it always should be possible to minimize actual memory consumption, by having the run-time system automatically load small modules of the executable program from their disk storage.

I ported a FORTRAN version of the popular game GOMOKU to the Amiga. It consists of a dozen subroutines, statically linked into a single 30 K executable image.

Simply by skipping the static linking step and letting the Absoft run-time system perform a dynamic linking I was able to reduce the permanent memory allocation for the main program unit to 7 K, without any further modifications. When run, the other program units were swapped in and out of memory as needed. They varied in size from about one to six K.

Naturally, there are some warnings when using the dynamic linking. The chief one is that unresolved references at compile time will not be discovered until run-time, since the compiler had been explicitly informed (via a compile time option) that the current module is known to be incomplete and will eventually be linked either statically or dynamically to complete the references.

A second price to pay is the extra time which is required to constantly load program code from a floppy disk. I do not consider either of these warnings particularly troublesome, since the former is easily handled by the debugging tool, while the latter may eventually be alleviated with by a hard disk.

Screen-oriented debugging tool

The debugger is an interactive, screen oriented tool for developing FORTRAN programs. With it, the values of program variables may be examined or changed, memory usage and logical unit connections displayed, and program execution controlled on an individual instruction basis with the aid of breakpoints.

Anyone who has ever toggled in a program by hand on one of those historic room-size personal computers allowing instruction-by-instruction progression will view the debugger as another great idea come home.

The debugger uses two overlapping windows; one to enter a battery of debugging commands and view their consequences, and one to accept another set of commands. Meanwhile, the FORTRAN source listing is displayed, with the current line of execution marked. The latter is much like the LIST window in Microsoft's Amiga Basic.

Both debugging windows have front-back and resizing gadgets, although movement between the windows is more efficiently done through the keyboard. Absoft claims that if you understand the use of the debugger, you should never have to imbed purely diagnostic PRINT or WRITE statements in your source code again. I agree completely.

There are, however, "bugs" in the debugger. One is the lack of cursor positioning information to keep pace with a resized window. Another problem, is the manual states that program input or output occurs at the bottom of the debugger window when in fact the actual I/O occurs in the CLI window from which the debugger was invoked.

These two problems can be overcome by keeping the debugging windows at their full extents, and using the front-back gadget to reach the CLI window where input/output is to be reported. In spite of these shortcomings, this tool is simply an indispensable part of the FORTRAN programming environment.

Distribution disk layout

The compiler, linker, library manager, and debugger are delivered on one disk, together with the executable version of the Absoft run-time library, and a small number of additional files necessary for the FORTRAN interface to C routines and to Amiga ROM kernel routines.

The disk is not copy-protected. These tools may be moved to other disks or directories. One restriction is that the run-time library must always be accessible to a compiled program, a feature which will surely limit the portability of some code.

There are four example programs, including a FORTRAN version of the "Hello World" program from the Intuition reference manual. These programs are specifically included because they demonstrate the use of include files, and the single FORTRAN subroutine call necessary to access ROM routines.

They also illustrate the use and acceptance (by the compiler) of structured programming features, an indication that these sample programs were actually coded by a diehard C programmer. Any old FORTRAN programmer who scrolls one of these listings by will surely get sea sick from the waxing and waning of all those tabbed-out statements!

There are no help files nor instruction files on the distribution disk. There is no on-line help in the form of pull-down menus in any of the above tools.

On the first day after I got the compiler, I did not actually try the supplied test programs nor read the reference manual with great detail. Instead, I wrote my own programs and plunged right in. I was delighted to find that my simple experiments were just as simple to compile with only the rudimentary knowledge gained from about an hour's worth of reading through the manual.

Nothing is perfect

In spite of my overall exuberance for Absoft's product, there were still some obvious flaws and certainly room for improvement. Most noteworthy was one piece of misinformation - an error message which said "out of memory" instead of "heap overflow". This prevented the compilation of the test programs provided on the supplied disk for almost a full day.

Perhaps nothing is so aggravating as being unable to execute a sample program specifically supplied for that purpose. By default, the compiler allocates a heap size of 20K, which is insufficient to compile these sample programs. The initial heap size is easily changed. A simple README.FIRST file could point that out the minimal heap for each sample program. Absoft has been apprised of this deficiency and will no doubt rectify the manual.

I also suggest that a Programmers Reference Card (about the size of a shirt pocket will do it) should be produced and included with the other documentation.

Even though the debugger has its own on-line help file, and the compiler and linker will prompt for their appropriate arguments in cases of user forgetfulness, this compendium of mnemonics is a familiar aid to most programmers, regardless of language.

Finally, the list price for this product (\$295.00) will probably be prohibitive for the occasional user of FORTRAN or for those users who would like to experiment with the Amiga as a workstation but are unsure as to its acceptability.

One suggestion would be to make available both professional and personal versions of the Absoft FORTRAN compiler at appropriate prices in a manner similar to marketing strategies adopted by other companies in this field.

AC/FORTRAN 77 Compiler and Debugger

List price \$250

Absoft Corporation

4268 N. Woodward
Royal Oak, Michigan 48072

•AC•

Using Fonts from AmigaBASIC, Part Two

... fonts & libraries and how to use them from AmigaBASIC

By Tim Jones
People Link AMSOFT !
SYSOP The Window BBS, (617)-868-1430

Libfunctions:

```
DECLARE FUNCTION AllocMem% LIBRARY
DECLARE FUNCTION OpenFont% LIBRARY
DECLARE FUNCTION OpenDiskFont% LIBRARY
DECLARE FUNCTION AvailFonts% LIBRARY
DECLARE FUNCTION AskSoftStyle% LIBRARY
DECLARE FUNCTION SetSoftStyle% LIBRARY
LIBRARY "diskfont.library"
LIBRARY "exec.library"
LIBRARY "graphics.library"
```

ScreenWindow:

```
WIDTH 78
```

SysVariables:

```
' All variables must be LONG if passed
' to functions
AFFMEMORY% = 1 ' find fonts in memory
AFFDISK% = 2 ' find fonts on disk
MEMFPUBLIC% = 0 ' memory must be fully public and
' relocatable
MEMFCLEAR% = 65537% ' presets memory allocated to zeros
BufSize% = 512 ' I want 1/2K for my buffer
' (10 bytes * #fonts + 2)
BufPtr% = 0 ' this is the pointer to the memory
' that was
' allocated by the AllocMem%() call
```

AllocateMem:

```
BufPtr% = AllocMem%(BufSize%, (MEMFPUBLIC% OR MEMFCLEAR%))
IF BufPtr% = 0 THEN
PRINT "Couldn't find"; BufSize%;
PRINT "Bytes of contiguous free RAM."
FreeMemExit
END IF
PRINT "Memory Allocation GOOD. "; BufSize%;
PRINT "bytes allocated. Searching for fonts..."
```

GetAvailFonts:

```
FontList% = AvailFonts%(BufPtr%, BufSize%, AFFDISK% OR
AFFMEMORY%)
IF FontList% <> 0 THEN
PRINT "Couldn't find Fonts. Cleaning up..."
FreeMemExit
END IF
PRINT "Found Fonts!!!"
PRINT "AvailFontsHeaders and AvailFonts structures created at";
PRINT BufPtr%
NumFonts% = PEEKW(BufPtr%)
PRINT "I found"; PEEKW(BufPtr%); "fonts on the disk and in
memory."
NewPtr% = BufPtr% + 4
```

GetFontNameAddresses:

```
DIM StrAddr%(NumFonts%)
FOR Num% = 1 TO NumFonts%
StrAddr%(Num%) = PEEKL(NewPtr%)
NewPtr% = NewPtr% + 10
NEXT Num%
```

GetFontType:

```
afPtr% = BufPtr% + 2
DIM afType%(NumFonts%)
FOR Num% = 1 TO NumFonts%
afType%(Num%) = PEEKW(afPtr%)
afPtr% = afPtr% + 10
NEXT Num%
```

GetThePointSizes:

```
DIM PointSize%(NumFonts%)
PointPtr% = BufPtr% + 8
FOR Num% = 1 TO NumFonts%
PointSize%(Num%) = PEEKW(PointPtr%)
PointPtr% = PointPtr% + 10
NEXT Num%
```

BuildNames:

```
DIM AvailName$(NumFonts%)
FOR Num% = 1 TO NumFonts%
Char$ = " "
WHILE Char$ <> CHR$(0)
Char$ = CHR$(PEEK(StrAddr%(Num%)))
IF Char$ <> CHR$(0) THEN
AvailName$(Num%) = AvailName$(Num%) + Char$
StrAddr%(Num%) = StrAddr%(Num%) + 1
END IF
WEND
NEXT Num%
```

DisplayNames:

```
CLS : COLOR 3,0
PRINT "Font Name"; SPACE$(7); "Pt Tp Font Name";
PRINT SPACE$(7); "Pt Tp Font Name"; SPACE$(7); "Pt Tp"
PRINT "-----"; SPACE$(7); "--- -- -----";
PRINT SPACE$(7); "--- -- -----"; SPACE$(7); "--- --"
COLOR 1,0
FOR Num% = 1 TO NumFonts%
StrLen% = LEN(AvailName$(Num%))
Pad% = 15 - StrLen%
YSize% = STR$(PointSize%(Num%))
IF LEN(YSize%) < 3 THEN YSize% = " " + YSize%
fType% = " " + STR$(afType%(Num%)) + " "
AvailName$(Num%) =
AvailName$(Num%) + SPACE$(Pad%) + YSize% + fType%
PRINT AvailName$(Num%);
NEXT Num%
```

New! Available thru Cardinal AMIGA™ SCHEMATICS

24⁹⁵

800-762-5645

(10-6 M-F)

- At last you can see for yourself about
- Memory Expansion
 - Enhancement
 - Added ports, etc.

Cardinal Software™



14840 Build America Dr.
Woodbridge, VA 22191
Info: 703-491-6494

Amiga is a™ for Commodore Business Machines.

GetDecision:

```
LOCATE 19,1 : COLOR 3,0
PRINT "Enter the name of the font and its point size";
Print ' from the chart above list"
PRINT "Type 'END,0,0' to exit";SPACE$(50)
PRINT "Font Name, Point size and Type(seperated by commas)"+;
Print SPACE$(20)
LOCATE 21,53 : INPUT Ft$,Pt$,Type$ : COLOR 1,0
IF UCASE$(Ft$) = "END" THEN
    FreeMemExit
END IF
IF UCASE$(RIGHT$(Ft$,5)) <> ".FONT" THEN Ft$ = Ft$ + ".font"
```

DisplayFont:

```
WHILE MOUSE(0) <> 0 : WEND
WINDOW 2,Ft$+" "+STR$(Pt$)+" Points", (0,0)-(631,186),0,-1
Rps = WINDOW(8)
Font Ft$,Pt$,0,0
enable$ = AskSoftStyles(Rps)
PRINT : COLOR 3,0
FOR i = 0 TO 7
    SetStyle CINT(i):PRINT Ft$+" "+STR$(Pt$)+STR$(i)+" ";
    SetStyle CINT(i + 8): PRINT Ft$+" "+STR$(Pt$)+STR$(i+8)
NEXT i
Type$ = 1
Font "topaz.font",8,0,0 : COLOR 1,0
PRINT "Click the MOUSE to continue"
WHILE MOUSE(0) = 0 : WEND
WINDOW CLOSE 2
GOTO GetDecision
```

SUB FreeMemExit STATIC

```
    SHARED BufPtr$,BufSize$,pFont$
    IF pFont$ <> 0 THEN CALL CloseFont(pFont$)
    IF BufPtr$ <> 0 THEN
        CALL FreeMem$(BufPtr$,BufSize$)
        CLS : PRINT : PRINT
        PRINT "Memory at";BufPtr$;"Returned to the HEAP."
    END IF
    LIBRARY CLOSE
    STOP
END SUB
```

SUB Font(fontName\$, height\$, style\$, prefs\$) STATIC

```
    SHARED pFont$,Rps,Type$
    IF pFont$ <> 0 THEN CALL CloseFont(pFont$)
    fontName0$ = fontName$ + CHR$(0)
    textAttr$(0) = SADD(fontName0$)
    textAttr$(1) = height$*65536$ + style$*256 + prefs$
    IF Type$ = 2 THEN
        pFont$ = OpenDiskFont$(VARPTR(textAttr$(0)))
        IF pFont$ <> 0 THEN
            CALL SetFont$(Rps,pFont$)
        ELSE
            PRINT "Couldn't set the font to ";fontName$
            FreeMemExit
        END IF
    ELSEIF Type$ = 1 THEN
        pFont$ = OpenFont$(VARPTR(textAttr$(0)))
        IF pFont$ <> 0 THEN
            CALL SetFont$(Rps,pFont$)
        ELSE
            PRINT "Couldn't set the font to ";fontName$
            FreeMemExit
        END IF
    END IF
END SUB
```

•AC•

AmigaNotes

"The Amiga sound situation is looking... er... sounding... better by the moment."

by Rick Rae
CIS# 76703,4253

The Amiga sound situation is looking... er... sounding... better by the moment. As of this writing (September), I have, in house, four MIDI interfaces, three audio samplers, and several software packages. And more goodies are on the way!

This month we have our first audio-oriented submission by another author. Stephen Pietrowicz shares his impressions of EA's Instant Music with us elsewhere in this issue (*see our special insert*). I received Stephen's article as I was working on MY review of Instant Music, so you'll get a double shot this month. Don't expect widely diverging opinions, though; we were both pleased with this product.

Before I launch into the review, I want to take a moment to make my first retraction. In the July issue I reviewed ActiVision's Music Studio. In that review I accused the authors of using "fairly large amplitude steps in their program", and blamed this for a certain roughness in the synthesized sounds as they swelled or faded out; I also complained about clicks and pops as voices were reassigned.

Since that review was written I have had a chance to play with several music and sampling programs, and ALL of them have exactly the same symptoms. I am now beginning to attribute these vagaries to the routines in the Amiga itself.

Still, I firmly believe that these extraneous noises can be eliminated through careful programming. We are just beginning our journey up the Amiga learning curve, and time will do wonders for the quality of our programs. What we already have is impressive and far outstrips what can be done on any other general purpose computer of comparable price.

REVIEW: ELECTRONIC ARTS INSTANT MUSIC

I wanted to start this review by classifying Instant Music, but I can't seem to find a category in which to place it, except for one: it's FUN. IM is different from anything which has gone before; it lets you create music, in real time, while knowing absolutely nothing about the subject. EA says the program uses artificial intelligence; I would say they are stretching the definition a bit. Let's take a look at exactly what makes IM work.

"Jamming" is a colloquial term for what happens when several musicians play together with nothing more than a song's framework in mind. I am involved with a very casual local band, and at least once during each practice session someone runs through

a chord progression or riff we all like. Within moments all the members are playing appropriate parts, and a tune springs to life. This is the essence of a jam: the spontaneity of the moment.

In order to jam, each player must have an understanding of the basic framework and rules of music; to simply play random notes is not sufficient. Perhaps the most fundamental rules are those of harmony.

For example, all musical tones can be described as ratios. Let's start with middle C. If we strike this key on a piano keyboard, a tone with a fundamental frequency of roughly 261.5 hertz rings out. If we strike the next C above, the tone created is roughly 523 hertz. This means that each "C" is twice the frequency of the one below it; in other words, the octave is a 2:1 ratio. If we strike both C keys simultaneously, the result is a very pleasant melding of the tones.

The next simplest ratio occurs if we strike the C and the G above it. This is referred to as a fifth, and is roughly a ratio of 3:2. Striking these two notes together is also a pleasing experience. But what if you strike a C and the B just below it? This is a rather distasteful combination, and it is because the ratio is roughly 18:17. Generally speaking, the simpler the ratio, the more pleasant the results. This principle, in conjunction with others, allows us to determine what notes will harmonize with a given melody.

Instant Music uses these rules to generate a "template" of acceptable notes, leaving the player the decision as to which of these notes will be heard. Although hardly in the category of artificial intelligence, it is a novel idea I haven't seen implemented elsewhere. It is vaguely reminiscent of the old "Music Minus One" records: the Amiga plays three backup parts and lets YOU play the lead line. The difference is, with Instant Music you can also pick which part you want to play AND be assured that you will always be on key.

Instant Music comes in a colorful "slip cover" folder which includes the disk, a 64 page manual, and 6 page reference card.

The disk is copy protected and utilizes a key disk scheme. It IS possible to make a backup using DISKCOPY, but you will have to insert the master disk briefly each time you boot. I am told that Marauder will make a bootable backup copy if the speed of the external drive is adjusted, but I'm not ready to tear my hardware down for this, so I can't verify the claim. If you munge your master disk EA will replace it at no charge during the 90 day warranty period, or for \$7.50 thereafter. To EA's credit they suggest you back up your master disk, and they do so on page one of the manual.

The Instant Music disk came to me write protected. This is an excellent idea: it protects you from yourself if you slip up while making the backup copy. I wish more manufacturers would flip or remove the write protect tab before shipment.

With the exception of the copy protection information, Instant Music is shipped on a stock AmigaDOS disk. It will run on a single drive 512K system and boots directly from 1.1 Kickstart; a second drive is helpful but not essential. For some reason Instant Music exhibited the same quirk as Music Studio: I was not able to launch it via CLI from my normal system disk without the Guru dropping by for tea and cookies.

The IM disk is not auto-booting; it loads up with a standard Workbench screen. To start the program, you must double click the disk icon, then the Instant Music icon, and then wait. And wait... the time from clicking the icon to a usable system is well over a minute. A part of this time is spent loading the introductory song, which begins playing automatically in mousejam mode.

One of the beauties of Instant Music is that you can start playing with it immediately. Grab the mouse, push the left button, and move it forward and backward: you're playing music! The motion of the mouse controls a small white cursor on the screen; this corresponds to the note to be played. When the left button is pushed, Instant Music uses its "template" and plays the closest selectable note at the next acceptable point.

An awful lot of time can pass while you do nothing more than mousejam using the default parameters and the supplied songs. Seven song subdirectories cover everything from classical to rock, with a grand total of 66 tunes and song segments. A few of these songs are VERY well done; my pick for best orchestration is Anitra's Dance.

GOING BEYOND THE DEFAULTS

Once you get the hang of mousejamming, you'll want to begin exploring variations on the stock themes. Many changes can be made on the fly as you play.

By pressing function keys F1 through F4 you can change the selected instrument -- the one you are mousejamming with - without having to stop playing. The up and down arrows allow you to transpose the song a half step at a time over a fairly wide range; again, this happens real time as you play. The right or left arrow will instantly return the song to the original pitch.

You can also change the rhythmic pattern in real time. Instant Music supports ten preset rhythms accessed by touching digits on the Amiga's numeric keypad. Or, you can turn the rhythm guides off completely by selecting Free mode. Between transposing, changing rhythms, and switching instruments, you can turn a simple repeating progression into a fairly impressive tune.

Instant Music gives you additional flexibility by allowing you to select the pitch template used. The Melody template is the most restrictive, sometimes allowing only selection of the octave in which a note is to be played. Chord mode gives you a bit more freedom, allowing you to select from the three or four notes which make up the selected chord; this lets you wander around quite a bit while still staying in perfect harmony. Scale mode opens up

the entire scale for the defined key; at this point you have seven notes per octave and can throw in a few klunkers if you aren't careful. Finally, Free mode turns the pitch template off completely, leaving you totally on your own.

After you've played with the default settings for a while, you'll probably feel very good about your skill at mousejamming. When your head starts to swell there's nothing like trying to jam with the pitch and rhythm guides set to "free" to bring you back down to Earth! I tend to forget just how much help the system is lending.

COMPOSING YOUR OWN SONGS

At some point you may want to try your hand at writing some of your own music, and Instant Music provides reasonable facilities for this. A song may be up to 64 measures in length, which is sufficient for most popular music and even some classical pieces.

There are no provisions for repeats, but the song automatically plays continuously. This is especially useful for rock and jazz jamming, where a given chord progression is repeated over and over.

There is no way to specify an intro or coda, so the entire song must be written out in its long form if this is required. You will also have to stop the song manually at its conclusion.

These limitations may sound restrictive, but remember that IM is not intended as a compositional tool. What it is supposed to do, it does quite well. In fact, the only thing that bothered me about composing was that I could find no way to clear or change a pitch guide. There is no way to start with a "clean slate"; you always begin with an existing song. Each song contains preset pitch templates, and I was never able to find a way to defeat or alter them. This is not a restriction if the pitch guide is set to Free during composing, but in mousejamming with the new song you may not play what you had intended!

One particularly nice feature is the cut and paste buffer. Instant Music allows you to mark any section of music and copy the music into the buffer; you may then paste the buffer into the song elsewhere. More importantly, you can cut a section from one song, load a new song, and then paste the buffer into the new song. This allows you to save files with snippets of songs and assemble them into a complete composition at a later date. This is especially useful with the drum kit: you can build up a library of patterns and then string those patterns together to form a complete song. This is the system used by most drum machines, and it works quite well.

SCREEN LAYOUT

Instant Music divides the screen into three sections. The uppermost section includes the menu bar and is used to select various options and control operating modes. Menus are used to load and save sounds, songs, and templates, edit the song in memory, and control jamming modes.

The bottom of the screen contains controls for the four instruments used in the current song. Each instrument has its own control block which indicates the name of the sample; the instrument which is currently selected (for mousejamming, sample loading, note placement, etc.) is highlighted.

Directly below each name is a slider gadget which controls the volume of the associated instrument. A cute trick is that the intensity of each instrument's color tracks the volume slider. Decrease the volume and the note blocks become dimmer; increase the volume and they become brighter. The only potential downside of this is that at very low volumes you can lose sight of the instrument altogether.

Also included is a display enable pushbutton for each instrument; click the button and that instrument's part is removed from the display. This is useful if you have a complex multi-part composition and want to look at only one instrument.

Finally, each instrument includes three buttons for selecting the octave in which it plays; this allows you to shift a sound up or down for the best rendition. More range would have been nice, but plus or minus one octave is certainly useful.

The majority of the screen is devoted to the song itself. This display takes the form of colored rectangles representing notes. Vertical position represents pitch, horizontal position indicates time, length is duration, and color indicates timbre (instrument). At first glance this seems to be the same system used by Music Studio on it's "musical paintbox" screen, but the resemblance is superficial. Music Studio uses an arbitrary relationship between block length and tone duration: one unit is a 32nd note, two units a 16th note, three units an 8th note, and so on. In contrast, Instant Music's block length is proportional to the tone: an 8th note's block is four times longer than a 32nd note, and the sounding of the notes corresponds exactly to the movement of the tempo pointer across the screen. This system is, to my mind, far superior to that used by Music Studio.

The screen is divided by a number of vertical lines which split the display into bars; these provide a handy reference as to where you are in the song. Along the bottom of the score display there is a stripe which is the rhythm line. At high magnifications this stripe displays the rhythm template; on more global screens it melds into a solid bar. Moving along this stripe during playback is a small pointer which indicates what portion is playing at each instant.

The program seems to be well thought out. Unlike many programs I've worked with lately, the requesters are very snappy and not sluggish at all. This may be due to the way the disk is broken down into small subdirectories. Whatever the technique, little touches like this show forethought by the author.

Also nice is the fact that everything is adjustable while the score is playing. You can change tempo, volume, you name it. In fact, you can even load and change instruments AS THE MUSIC CONTINUES. The only clue that something special is going on is an occasional break in the tempo of the song. Nicely done to say the least.

SAMPLED SOUNDS

Electronic Arts didn't skimp with the selection of instruments; there are 19 on the disk, running the gamut from the traditional piano and strings to esoteric sounds like DoVoice (a person singing "Do") and synthesized textures. Only a few of these sounds are so bad as to be pointless, and some of them are excellent. The only negative point is that most are not multi-

VT100 Emulation

for the
Commodore Amiga™

MiddleMan™

**Communicate with information services,
mainframes, and other personal computers**

- Provides true DEC™ VT100™ emulation
- Emulates the VT100 numeric keypad
- Supports uploading and downloading of text files
- Supports baud rates from 300 to 19200
- Displays text at a rate of approximately 7000 baud
- Provides eight user definable macros
- Uses a four color text screen
- Executes comfortably on a 256K Amiga
- Utilizes the full capabilities of the Amiga graphical interface and multi-tasking operating system
- Upgrade plans include Tektronix® 4105 color graphics terminal emulation

Available Now - only \$59.95
Benalah Computer Products, Inc.
 P.O. Box 11165
 Huntsville, AL 35814-1165
 (205)859-9487

sampled; in other words, one sample is made to stretch across the entire musical range. Due to the laws of musical acoustics this results in a sound which is only realistic across a narrow range. (We'll be taking a look at this phenomenon in a future column.)

The Drum Kit voice is particularly interesting, because it IS multi-sampled and includes five instruments: bass drum, tom tom, snare drum, high hat cymbal, and wood block. There are eight tunings for the wood block and twelve for each of the others.

Of course there is nothing to prevent you from importing your own instruments, or even creating your own multi-instrument sets like the Drum Kit. All of the samplers I have at this time are capable of writing IFF format files, and any of them may be used to create instruments for IM. (A review of these samplers is coming up; stay tuned!) I tried my hand at this almost immediately; after all, what good is a DoVoice without a WopVoice? <Grin>

THE INSTANT MUSIC MANUAL

The manual is very well done, and is divided into several logical sections. You can read far enough to learn what you need to know and ignore the rest with very few side effects.

The first section is a quick introduction to the program, including information on mousejamming, loading songs and instruments, changing existing songs, and saving modified scores. This section, ten pages in length, is all you need if you simply want to make music with what is provided.

The second section goes into greater detail, providing information on creating your own songs and templates. This section is divided into tiny lessons called "etudes" (technically, a piece of music taught for the sake of learning a lesson other than the music itself). Each etude covers a small part of composition, such as laying down bass lines, placing chords, or the like. I would have enjoyed a few examples which went through the complete composition of a song from beginning to end, but the information provided is sufficient.

The third section is a reference which defines the terms used by Instant Music and explains the function of all commands. Following this is a glossary of musical terms which covers a fairly wide spectrum. Probably very few of us need to have a chord explained to us, and even fewer want to know anything about Mixolydian mode, but it's all there for the perusing.

The fourth section is an appendix which comments on all the songs included. This is a handy section for study, as it details the progression and rhythmic devices used in each song and suggests modifications for the user to try. This is especially useful for the tunes in the Progressions subdirectory, which are chord changes taken from various rock and jazz tunes. These progressions are excellent starting points for your own songs.

The manual seems to be geared to a one drive system. It mentions that there is room for about forty songs per backup disk and suggests that you make multiple backups for additional storage space. Since the requester supports DF1 I see no reason not to use data disks for virtually unlimited storage if you have a second drive. Speaking of this, one of the nice features for the future is support for HD1. This was ghosted in my copy, presumably because I don't have a hard disk installed.

Here once again is a manual with no index, but somehow I don't mind this one as much as usual. The table of contents is done in outline fashion, making it relatively easy to locate a desired section.

FLAKIES

I ran across a few minor bugs in both the program and the manual. Fortunately most were not fatal, and all are avoidable once you know about them.

The manual suggests that you back up the master disk, and warns you to name the backup disk anything but "IM", which is reserved for the master disk. Fine, thought I, let's do that, and we'll call the backup "InstantMusic". The backup proceeded without a hitch, but the disk refused to boot: I would get an infinite pause after the Instant Music logo screen, and double clicking would simply restart the boot with the same result. The reason? The master disk is NOT named "IM" as the manual indicates; instead it's named -- you guessed it -- "InstantMusic". Name your backup anything else and you'll be safe.

When playing in mousejam mode, the lowest note is sometimes enabled when it shouldn't be: dragging the cursor block to the absolute bottom of the screen plays notes whether they are in key or not.

Moving the tempo bar all the way to the left sometimes sets zero tempo: the playback freezes until the slider is moved to the right.

On some disk requesters there is blank space below the last file name. If you click in this area, the requester will set up a file name which is a random truncation of one of the listed names. (If you should be so silly as to try to load this file, IM catches the error gracefully and posts a "File not found" message. You would never do this under normal circumstances; remember, I was TRYING to crash the system!)

I did a lot of menu and option manipulation while songs were playing, and occasionally I confused the system as it was refreshing the screen, resulting in random diagonal lines being drawn. Some went away, others remained on the screen until I reset the computer.

When experimenting with imported samples, I was able to crash Instant Music by playing large samples. A 42K sample created with the Mimetics sampling package caused playback to freeze after playing only a few notes; the only cure was a reboot. An 18K sample created with the same software loaded and played correctly, so I would guess the maximum sample length to be 32K. Apparently the author did not anticipate the possibility of importing extremely large samples.

IN CONCLUSION

Instant Music will probably go in my bottom drawer after this review is written, since it's a program I won't be using very often. On the other hand, when people come over and I want to impress them with my Amiga, IM will probably be one of the first programs I run. And that, my friends, is one of the highest compliments I can pay to a program.

That's all for now. Till next month...

Nybbles,
Rick

SUMMARY: ELECTRONIC ARTS INSTANT MUSIC

This one is especially for those of us with little or no musical skill. I found a few bugs, but most of them are non-fatal and all are avoidable. At the price you should definitely consider it if you are interested in music at all. For the sake of fun, a hearty thumbs up.

PRODUCT: Instant Music \$49.95

COPY PROTECTION: Key disk

REQUIREMENTS: Amiga with 512K, one drive, KS 1.1

Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
800-245-4525

•AC•

Roomers

The most-asked questions of a rumor-monger and more.....

By John Foust

Fill in the blank: When will _____ arrive? This is the most-asked question of a rumor-monger. The most popular blanks are:

- A. Sidecar
- B. AmigaDOS 1.2
- C. Genlock
- D. Flight Simulator
- E. Turbo Pascal

followed by generic pleas for 'a good word processor', 'a good database' and 'a good programming language'.

And what might be the answers to these questions?

A. Sidecar: Perhaps mid-December. Your local dealer will probably get a demo unit several weeks before you can buy one, for between \$695 and \$795. Production was rumored to begin in early November, but one rumor persists that there is a warehouse with 1,500 Sidecars here in the US.

B. You might have it already. When I wrote this, the master disks had not been duplicated, and only developers had 1.2 gamma 1 release disks. More details can be found below.

C. Genlock is being manufactured now, according to a Commodore-Amiga engineer who worked closely with the design. Some delays are expected because of problems finding second sources for some electronic parts in the Genlock.

D. Again, you might have it now. Working demos were shown at the West Coast Commodore Association convention. According to some, Jet will be released two months after Flight Simulator. I presume its release also depends on the sales of Flight Simulator.

E. Even the legendary Borland has fallen to advertising non products. They are making enough money in the IBM P C world to afford large ads in AmigaWorld, and to whip up the wrath of a group of computer owners as large as the Amiga market. Some people sent letters to Phillippe Kahn, the head of Borland. His replies said "No way," in so many words.

Of course, this leaves the possibility that another company will out-Borland Borland, and produce a nice interpretive Pascal for the Amiga. Of course, Borland could port Turbo Pascal for much less money and effort than someone else could write another Turbo Pascal.

AmigaDOS 1.2 nears shipment

AmigaDOS 1.2 will be shipped to dealers in late November. It is expected to retail for \$15. It includes three disks, and a manual printed update for the new commands, in an Amiga-style binder, with tabs on the side.

The Extras disk includes a utility that reads and writes IBM format 5 1/4 inch disks under AmigaDOS. It should also have an update of Amiga Basic, with a few new commands, but mostly bug fixes and speedups.

AmigaDOS 1.2 developer updates

AmigaDOS version 1.2, gamma 1, number 33.44 was shipped to developers in mid October. It was a seven disk set.

Some developers have reported strange lock-ups with this version, especially when initializing disks. Hopefully, the release version will be truly free of strange bugs.

This was the last free update to today's developers. Commodore-Amiga will be switching to a new developer support plan. Future updates will be available for a price. The details of this plan haven't been released yet, but the bug report sheets for the gamma 1 release offer a dollar credit reward to developers who report true bugs in this version.

Big surprises

The biggest surprise in the gamma update was the Wack documentation. Wack, a debugger, was supplied on the first developer tool disks without documentation. It has more features than any developer ever imagined. It was just sitting on our developer's extra disk, all this time.

One night, when I had a little free time, I looked through the binary code, using the CLI 'type opt h' command. (The words 'free time' on my lips means I'm talking about pre-Amazing Computing days.) I saw many debugging command keywords, and I couldn't guess at their syntax or use, so the disk went back in the box. Other developers surely did the same thing.

The real Wack has a Lisp-like debugging language that allows you to write extensible macros that are executed at specified conditions, and things like that. Now developers can write Lisp-like macros to poke around in BCPL and 68000 assembler, and hunt bugs in C compilers. Somehow, this all makes sense. It's not just a game to force you to learn more than one language. Really.

Infominder demo

The gamma disk set contained a demo version of Infominder, the information organizer from Jim Becker of Terrapin Software, distributed through Byte-by-Byte. The program was loaded with an index to the 'autodocs', the self-documentation traditionally supplied with the developer updates.

Why did these disks carry a warning "Do not duplicate without express written permission"? Early in the summer, Commodore sent developers the 1.2 beta 4 disks. Soon afterwards, developers gave copies of these disks to their local dealer, and then they spread to local user groups.

In effect, if not in deed, Commodore had just released a new buggy operating system. So Commodore sent a memo to all dealers, reminding them not to copy the beta disks for customers. Of course, it wasn't an official release. But tell that to the press. Computer Chronicles and CompuServe's Electronic Online Today reported that Commodore was recalling its buggy operating system.

New Lattice compiler 3.1

It is said that the developers are getting a not-so-great Lattice C compiler in the next update. One version will be given to developers, and another more optimized and feature-packed version will be sold separately. The next version of Lattice C is 3.10.

Developer Conference

It is hard to guess what will happen there, since the agenda was circulated only a week or so before the conference. Developers

jumped at the chance to go, without any confirmation of the content of the conference. It could have been about Amiga Basic, for all anyone really knew.

Tuesday is registration and a cocktail party. Wednesday will have Carl Sassenrath, the author of Exec talking about the Exec, a speech on the ANSI C standardization effort, someone from Electronic Arts will talk about workstation development environments, (Hmm, a lot of developers have those in their garage...) a talk on IFF by Commodore West Chester techie Carolyn Scheppner, Los Gatos' Jim Mackraz on the improvements to AmigaDOS 1.2, RJ Mical on a programmer's suite of high-level routines to make Intuition even easier, Dale Luck on graphics, and Barry Whitebrook on advanced graphics topics.

Thursday has Neil Katin and Glenn Keller on bus expansions and interfaces, a talk about MIDI and Amiga music, Gail Wellington on European topics, Tim Jenison, of New Tek on video processing, a talk on scientific applications of the Amiga. Last but not least, a representative from marketing will uncover the highly secret future products in the Amiga line.

Friday is the closing remarks, a press conference, and a developer's fair, open to the public. Several workshops will be held in the afternoon, including a presentation from Lattice, Luck and Mical on Intuition and graphics, another MIDI presentation. Finally, Jom Goodnow, of Manx Software, will demonstrate his new debuggers. The evening will linger over an awards dinner.

•AC•

A programming environment
designed specifically for the Amiga™.

Multi-Forth™ for the Amiga.



Multi-Forth is a new language which was designed to unleash the full power of the Amiga. Multi-Forth provides complete access to all Amiga libraries including Intuition. It compiles stand-alone applications in seconds (other languages typically take several minutes). There are no royalties, and no "levels." CSI provides the best support of any computer language vendor, including CSI technical hot line, our own CompuServe net (GO FORTH), and comprehensive documentation. Programming the amazing Amiga is interactive and fun with Multi-Forth. Contact us for a technical data sheet with the complete list of Multi-Forth's features.

**Simply the best programming
environment for the Amiga.
\$179 Shipping Now.**

Multi-Forth is a trademark of Creative Solutions, Inc.
Amiga is a trademark of Commodore-Amiga, Inc.

Creative Solutions, Inc.
4701 Randolph Road Suite 12
Rockville, MD 20852
(301) 984-0262 in MD or
1-800-FORTHOK

Forth!

by Jon Bryan

I'd like to think that some of you readers out there are enjoying my series of articles on the language called "Forth". No, I take that back. I hope that a LOT of you are enjoying it! Things have been a little slow in my first few installments, and I apologize for that. At first there wasn't much that I could do simply because I didn't have anything to do it with. Multi-Fort from Creative Solutions solved that problem, and I've since shown you an implementation of a circle-drawing algorithm and some sprite tools. They demonstrated some of the unique characteristics of Forth in general and Creative Solutions' implementation of the language in particular, but were still pretty limited.

One of my goals in writing these articles is to illustrate the power, utility and flexibility of this truly unique language. As a vehicle for the task, I proposed back in the July issue that I would write a three-dimensional simulation of a bouncing ball. Well, without further fanfare, here it is!

HOW DO YOU DESCRIBE A BOUNCING BALL?

If you were given the task of describing, in English, exactly how a ball bounces, what would you say? That it starts with some initial velocity and flies through the air until it hits something? That it then bounces? That it loses a little bit of energy on each bounce, and that air friction works to slow it down some more? Oh, and don't forget gravity.

So, after bouncing around for a while the ball's velocity eventually decays to nothing and it stops, right? Well, as we all know, computers need precise instructions. A program to animate an image of a bouncing ball on a CRT has to be very precise.

Precisely what I've done in this demo program is this: first, a custom screen and window are opened, the perspective view is drawn in the window, and the ball sprite is set up. The main loop animates the ball. When the velocities decay to zero the loop starts over with new velocities. Within the loop the window is monitored for a click on the close gadget, which causes the sprites to be freed and the screen closed. Here's how it looks in Forth:

```
: Bouncer  ( -- )
  Initialize
  BEGIN InitVelocities
    BEGIN
      BouncerEvents DoMove Bounce Stopped?
    UNTIL
  AGAIN ;
```

Rather concisely put, if I do say so myself.

BACK TO THE BEGINNING

All the details of the code are hidden in a few simple words. Let me go through the listing step by step and explain them.

But first, let me say that you're wasting your time trying to do any serious programming on the Amiga without the full set of

reference manuals. I would also like to commend the people at Amiga who put them together. Bravo!

The Hardware Reference Manual gives all the details on creating, changing and moving sprites, along with a tremendous amount of other information. One of the things it says is that "it is convenient to lay out the sprite on paper first." I don't know about you, but I'd rather let the computer do the work.

The first few words in the listing are an implementation of some sprite-defining tools. In fact, they're basically the same tools that were in my last column. I've made some changes (dare I say improvements?) after thinking about the problem for a few more weeks, but the basic idea is still the same. The word `Sprite` interprets a block of ASCII characters and lays down the binary image of a simple sprite in memory. Attached does the same for an "attached" sprite. They provide simple, but powerful, tools for the creation of sprite images.

The real power of Forth can be seen in the word `MakeBall`, though. `MakeBall` is something Forth programmers call a "defining" word. It allows the creation of a class of words which all share the same run-time behavior. The sprites named `0Ball`, `1Ball`, `2Ball`, `3Ball`, `4Ball` and `5Ball`, when executed, `CHANGE THEMSELVES!` That allows an elegant solution to the problem of changing the size of the ball as it moves further away from and closer to our point of view.

After the definitions of the ball sprites are the following two lines:

```
CREATE BallVectors
] 0Ball 1Ball 2Ball 3Ball 4Ball 5Ball [
```

For those non-Forth programmers out there, `CREATE` builds the header for the word `BallVectors`, but doesn't allocate any space in the dictionary. When executed, `BallVectors` will place its address on the stack, which in this case will be the address of the first value in an array of execution vectors. The Forth word `]` (right bracket) turns on the compiler and causes the 16-bit tokens for the words which follow it to be placed in the dictionary. The word `[` (left bracket) turns the compiler back off. The word `ChangeBall` takes a number off the stack, multiplies it by two to get a word offset which it adds to the address provided by `BallVectors`, and fetches the appropriate token for subsequent execution. Later, when we want to change the size of the ball all we need to do is calculate the vector.

If you're not a Forth programmer you're probably saying "what is this turning the compiler on and off?" That's an excellent question. Forth doesn't "compile" in the same sense that C or Pascal does. What it does is find words in the dictionary and either execute them or store their address or token (depending on the implementation) in the "parameter field" of the word being defined. The word `:` is a good example. When "colon" executes it looks forward in the input stream for a name and lays down that

Ami Project

Journal for the Amiga Computer

Are you tired of Amiga magazines that seem to be written for beginning programmers? Do you already know how to program, but want to understand the Amiga? Ami Project is a monthly journal dedicated to bringing you the techniques and examples for using all the power contained in your Amiga computer. Intuition, Bobs, VSprites, IFF, Multitasking, are all covered in depth within our pages. Are you ready for some real Amiga programming? Subscribe today!!!

\$24⁰⁰ For 12 issues (1 year)

Send Check or Money Order to

Nortia
P.O.Box 285
Kent, OH. 44240

Canada \$30 U.S.
Dealer Inquiries invited
(Formerly Amiga Project)

sequence of characters as part of the "header" for a new word. This program takes about twenty seconds to compile, so you can see that it's a very efficient method. (To complicate things a bit, some words are "immediate" and execute even when the compiler is turned on. Then there are the "subroutine-threaded" Forths, which really DO compile machine code for each word. Sorry.)

Continuing with bouncing balls, the words which are used to define the shadows are much the same as those for the balls, but the shadows are "simple" sprites, unlike the "attached" ball sprites.

After the sprite definitions are the words for getting the sprites from the operating system and freeing them when the program finishes. GetBall tries to get two consecutive sprites and aborts with an error message if it's unsuccessful. GetShadow will only accept the sprite it asks for, number seven, because it requires specific color registers. Of course, if one of the requests gets a sprite that it can't use, it has to free that sprite before it can continue.

Moving the ball and shadow are accomplished at the lowest level by MoveBallSprite and MoveShadowSprite which expect screen coordinates on the stack and make calls to the system routine MoveSprite.

The colors for the shaded balls are set by 19-31. Greys, which explicitly sets color registers 19 through 31. There's a little story in this. I began by assigning a brighter shade to each successive register. Then when I started work on the shadow, I discovered that I couldn't make it a dark color without assigning it to a lower-

numbered sprite than the ball sprite. Because a low-numbered sprite has a higher priority than one with a higher number, that meant the shadow appeared in front of the ball when it was rolling on the floor. I had to go back and redo the sprite definitions.

A DIFFERENT PERSPECTIVE

We're finally to the portion of the program that animates the ball. First there are a number of constants and a few variables. These define the coordinate system that the ball will move within and provide the proportions to generate the proper perspective view. There are constants for the force of gravity and the coefficient of restitution for the ball, and variables which will hold the velocities in each axis. The Z-axis (depth) coordinate is also kept in a variable to cut down on stack manipulation.

The word Perspective generates the screen coordinates for a given X, Y and Z axis position. Ycrt and Xcrt use it, whereas Zcrt simply divides the Z-axis position by a constant (4096). The value (0-5) returned by Zcrt is used later by ChangeBall to change the size of the ball and shadow.

MoveBall puts all the perspective calculations together. After the screen coordinates and size vectors are calculated for both the ball and its shadow, the system routine WaitTOF waits for the vertical blanking period before moving the images. If the sprites were moved while the beam was drawing the screen display they would be "sliced" up frequently because they would move somewhere else before they had been completely drawn. I had to write my own routine to wait for blanking when I did this on the Commodore 64.

There are several words which clip the X, Y and Z coordinates in order to keep the ball within the confines of the "room." There is also a subtle problem to be solved here. Suppose the ball is moving very fast and when the next position is calculated it's outside the room. We can clip the coordinates to the boundaries, but that's only part of the problem. Suppose the ball is falling and its velocity is increasing. If the new velocity is already calculated and we simply clip to the boundaries, the ball gains velocity that it shouldn't have, allowing oscillatory conditions to occur. In other words, the ball can get stuck bouncing and never stop (which is a bit unrealistic). I know because I did it wrong the first time.

The solution I use is to test the coordinates and, if they're outside the room, to calculate the actual velocity using the equation:

$$+-\text{SQRT}(\text{Calculated_Velocity}^2 - (2 * \text{Acceleration} * \text{Distance_outside}))$$

-YvelAdjust, YvelAdjust, and AdjustVelocity are the RPN implementation of the equation. You'll notice that they only deal with the Y axis. The X and Z axis don't have the problem. I won't go into the derivation of the equation, but it comes from the equations that follow.

We're gradually working up to gravity, but first let's take a look at friction. There are several variables for the remainders of divisions, and constants for air and surface friction which will be stored in the variable FrictionCoef. In the word Friction, I do some fixed-point scaling. The problem is that with integer values calculating a small percentage has to be handled specially for small values. I've set AIR to 999 and Surface to 990, which will

provide a coefficient for air friction of 0.1% and surface friction of 1.0%. If I did a simple

`Xvel @ FrictionCoef @ 1000 */`

for example, with a value for FrictionCoef of 999 and an Xvel of 10, I'd get an answer of 9 instead of 9.99. The solution is to scale Xvel up by 1000 before the calculation, then scale it back down using 1000 /MOD to get a quotient and a remainder and save the remainder for use the next time around. It gives three decimal places of accuracy to the calculation and prevents the velocity from decaying unnaturally quickly.

And now, Gravity. Here are the equations.

`Distance=Initial_Velocity*Time+Acceleration/2*Time^2`

`Final_Velocity=Initial_Velocity+Acceleration*Time`

We can arbitrarily, and conveniently, say that one unit of time passes each time we move the ball. Then, with Time=1, the equations simplify to:

`Distance=Initial_Velocity+Acceleration/2`
and

`Final_Velocity=Initial_Velocity+Acceleration`

assuming cancellation of units. Since the value for acceleration is constant, Acceleration/2 is a constant as well, allowing the calculations to be reduced to two additions. NewX, NewY and NewZ calculate the new coordinates and velocities with the appropriate adjustments for gravity and friction, and DoMove puts it all together.

The next thing you'll see is the word Blip, which does nothing at all. Next month I intend to give you the extensions to add sound. I simply ran out of time this time around, and I apologize.

With movement and sound out of the way the next thing to handle is the bounce. I've arbitrarily set the coefficient of restitution of the ball at 95%. Each time the ball strikes a surface it will lose 5% of its velocity in the axis perpendicular to that surface.

With clipping already in place, all that remains for the detection of a bounce is a test for equality with the maximum or minimum value on each axis. Or is it? Say the ball is on the floor rolling around. Going back to the sound which isn't done yet, if the only condition for a bounce (which assumes a Blip) that is tested is equality with the maximum value of Y we'll get some very strange sound effects for a rolling ball. (It sounded like a motorboat on my C-64.) The word Enough? checks to see if the ball has enough velocity to make noise. Finally, if the ball is determined to be rolling, the friction coefficient is changed. At first I thought that changing the friction coefficient would be a problem, but it turned out that the test for changing it came free as part the bounce.

A perspective view is the next requirement. The ball wouldn't bounce very realistically against a blank screen. DrawBackground uses Multi-Forth's graphics extensions moveto and drawto in conjunction with the system call SetApen to draw a simple perspective view of the room using the border color.

Before the view can be drawn a window has to be opened for it, and before that can be done a screen has to be opened. The

THE BLACKJACK



**Simulator
for the AMIGA 512K
Only \$24.95**

Now play **BLACKJACK** on your AMIGA just like you were in Nevada. Deals up to nine players using a simulated shuffled deck. The program actually analyses and reports on your progress during the game so you can mathematically build your own system of betting and winning. With Hi-res graphics and color, it's like you're playing with genuine cards. The instructions are built into the program so there are no manuals to lose. **BLACKJACK** is educationally recommended for helping children with their addition.

Dealer inquiries invited. Free shipping in the U.S.
Send \$24.95 to:

THE SOFTWARE FACTORY
4574 Linda Vista
Boise, ID 83704
(208) 322-4958

BounceScreen and BounceWindow structures are used for those calls. To avoid recalculating the view and keeping track of offsets I made the window immovable and unsizeable. The window is positioned a few lines down from the top of the screen to expose the screen's drag bar and allow it to be moved (try it!). The only event I'm looking for is a click on the close gadget in the window. Since we must be polite when programming on the Amiga, there are a few words which free the sprites and close the window and screen at exit.

Unless we want the ball to bounce forever a test for clicks must be made within the main loop. The word BouncerEvents checks for a CLOSEWINDOW message. The CASE statement is used even though an IF ... THEN would do just as well because other tests might be added in the future.

In the home stretch now, Initialize opens the screen and window, gets the sprites and sets their colors, draws the background, and places the initial values for the X and Y axis positions on the stack. InitVelocities sets the friction coefficient, zeroes the velocity remainders, and selects the initial X, Y and Z velocities.

You'll see the definitions tst0 and tst1 at the very end of the listing. I decided to leave them in to let you see one of the methods I used for debugging. The first moves the ball slowly from front to back. I used it to fudge the constants and correct the room proportions. The second displays the ball in one position for the same purpose.

A FEW FINAL NOTES

There are endless improvements that can (and will) be made to this program. Obviously sound remains to be implemented. I'd

like to add a control panel with slider gadgets for varying the coefficients, or use the mouse for a throttle. I expect other versions of Forth to be out soon and I'll be translating it over to those dialects. I may discover a much better way to do something that I would want to share with you. It should make an excellent vehicle for illustrating the Amiga's extensive capabilities.

All that aside, I would like to know whether you're getting anything out of this column. Are you enjoying it? What are your suggestions? My User I.D. on CompuServe is 73557,465 and on PeopleLink it's JON*FORTH. Feel free to leave a message. I post all of the code from this column to the Forth forum on CompuServe (GO FORTH, DL3), and the "turnkeyed" version of this month's program that can be run from the CLI will be downloadable from PeopleLink, and probably CompuServe as well.

Have fun with this machine! I know I am!

```
\ This is an example of hardware sprite animation
\ using attached sprites.
\ Jon Bryan:10-16-86
```

```
anew DemoMarker
\ If DemoMarker exists, it and all subsequent words are
\ forgotten and a new word DemoMarker is then created
\ which does nothing. Handy during development.
```

```
DECIMAL
256 CONSTANT ScanBufSize
CREATE ScanBuf ScanBufSize ALLOT
```

```
: SpriteLine ( -- addr1\addr2 )
  ScanBuf ScanBufSize INFILE @ READ.TEXT 1- ( trim delim )
  ScanBuf + DUP 16 - ;
```

```
: ?SpritePixel ( character\base -- value )
  DIGIT NOT ERROR" Illegal Sprite Color" ;
```

```
: OR_SpritePlanes ( number\address -- )
  SWAP 2 /MOD ( separate the two bits)
  SWAP 16 SCALE ( slide the low-order bit up a word)
  OR ( put them back together)
  OVER @ 2* ( move the stored value one place left)
  OR SWAP ! ( and OR the new bits into place.) ;
```

```
: DoSimplePlanes ( image\height -- )
  0 DO SpriteLine
    DO IC@ 4 ?SpritePixel OVER OR_SpritePlanes
  LOOP 4+
  LOOP DROP ;
```

```
: ImageSize ( height -- height\#bytes ) DUP 4* 8+ ;
```

```
: Sprite ( height -- )
  ImageSize
  CREATE HERE
  LOCALS| image size height |
  size ALLOT image size ERASE
  image 4+ height DoSimplePlanes ;
```

```
structure AttachedSprite
  simpleSprite STRUCT: +asEvenSprite
  simpleSprite STRUCT: +asOddSprite
structure.end
```

```
: OR_AttachedPlanes ( char\even sprite\odd sprite -- )
  LOCALS| odd even |
  DUP 4/ odd OR_SpritePlanes \ shift the two MSB's
  3 AND even OR_SpritePlanes ; \ mask the two lowest bits
```

```
: aImageSize ( height -- height\offset\total size )
  ImageSize DUP 2* ; \ for two sprites
```

```
: DoAttachedPlanes ( image\height\offset -- )
  LOCALS| offset |
```

```
0 DO SpriteLine
  DO IC@ 16 ?SpritePixel \ allows characters 0-F
  OVER DUP offset + OR_AttachedPlanes
  LOOP 4+ \ increment the pointer
  LOOP DROP ;
```

```
: Attached ( height -- )
  aImageSize
  CREATE HERE
  LOCALS| image size offset height |
  offset 2+ W, \ lay down offset to "attached" image
  size ALLOT image 2+ size ERASE \ reserve the space
  128 image 2+ offset + ! \ set "attach" bit
  image 6+ height offset DoAttachedPlanes ;
```

```
: +EvenImage ( addr1 -- addr2 ) 2+ ;
```

```
: +OddImage ( addr1 -- addr2 ) DUP W@ + ;
```

```
struct AttachedSprite Ball
  15 Ball +asEvenSprite +ssHeight W!
  15 Ball +asOddSprite +ssHeight W!
```

```
structend
```

```
: MakeBall ( height -- )
  Attached \ CREATE is imbedded here
  DOES> ViewAddress +vViewPort @ SWAP 2DUP
  Ball +asEvenSprite SWAP +EvenImage ChangeSprite
  Ball +asOddSprite SWAP +OddImage ChangeSprite ;
```

\ The values for the following images were derived with a
 \ combination of an equation gleaned from "Graphics and Image
 \ Processing" by Theo Pavlidis and "Calibrated Eyeball."

```
15 MakeBall 0Ball
00000007777000000
0000754444570000
00A6544334456A00
0086544334456800
0B876544445678B0
0B987665566789B0
ECA9877777789ACE
EDBA99888899ABDE
EEDCBAAAABBCDEE
0FEEDCCCCCDEEF0
0FFEEEEEEEEEEFF0
00FFEEEEEEEEFF00
00FFFFFFF0000000
0000FFFFFFF00000
000000FFFF000000
```

```
15 MakeBall 1Ball
00000000000000000
00000877778000000
0009544334459000
00A6544334456A00
0097654444567900
0B987665566789B0
0DA9877777789AD0
0EBA99888899ABE0
0EDCBAAAABBCDE0
0FEEDCCCCCDEEF0
00FFEEEEEEEEFF00
00FFFFFFF0000000
0000FFFFFFF00000
0000000000000000
```

```
15 MakeBall 2Ball
00000000000000000
00000000000000000
00000097790000000
0000964334690000
0009654334569000
0009766556679000
00B9877777789B00
00CA99888899AC00
00ECBAAAABBCDE00
000EDDCCCCDE000
000FFEEEEEEEEFF00
0000FFFFFFF00000
000000FFFF000000
0000000000000000
0000000000000000
```



```

15 MakeBall 3Ball
0000000000000000
0000000000000000
0000000000000000
0000009779000000
0000A743347A0000
0000965445690000
000A87777778A000
000CA988889AC000
000EDBAAAABDE000
0000EEDDEEE0000
0000FFFFFFF0000
000000FFFF000000
0000000000000000
0000000000000000
0000000000000000

```

```

15 MakeBall 4Ball
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000009669000000
0000075335700000
0000976556790000
0000B987789B0000
0000EDBBBDE0000
00000FEEEF00000
000000FFFF00000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000

```

```

15 MakeBall 5Ball
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000009669000000
0000096336900000
00000B9779B00000
0000EDCCDE00000
000000FFFF00000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000

```

```

CREATE BallVectors
] 0Ball 1Ball 2Ball 3Ball 4Ball 5Ball [

\ For non-Forth people:
\ The compiled form of a word in Multi-Forth is a 16-bit
\ "token." The ] turns on the compiler and [ turns it off.
\ The result is that six 16-bit values are stored
\ consecutively in memory. "BallVectors" puts the address
\ of the beginning of the array on the stack.

: ChangeBall ( n -- ) 2* BallVectors + W@EXECUTE ;

struct SimpleSprite Shadow
18 Shadow +ssHeight W!
structend

: MakeShadow ( height -- )
Sprite
DOES> ViewAddress +vViewPort @ Shadow ROT ChangeSprite ;

\ These simple sprites are a bit taller than the ball sprites.
\ That way they both use the same x,y coordinates and no offsets
\ are necessary.

```

```

18 MakeShadow 0Shadow
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000

```

KILL A MOUSE GO TO JAIL!

**MOUSEWASH IS A SPECIALLY
DESIGNED BALL WHICH:**

- CLEANS THE INSIDES OF YOUR MOUSE!
- SAVES YOU TIME AND MONEY!
- CAN BE USED HUNDREDS OF TIMES!
- NEEDS NO CHEMICALS

**SO TREAT YOUR MOUSE
TO MOUSE WASH TODAY**

T & L Gallery Vol. #1 \$19.95
OVER 85 IMAGES FOR DELUXE PRINT™

LET SAYIT! READ YOUR TEXT FILES TO YOU!
IT READS MOST STANDARD TEXT FILES ALOUD!

MOUSE WASH \$7.95--SAYIT \$14.95
\$2.00 SHIPPING \$3.00 FOR C.O.D.

T & L PRODUCTS
2645 Wilson Street
Carlsbad, CA 92008 (619) 729-4020
™ of Electronic Arts

```

0000000000000000
0000000000000000
0000222222220000
022222222222220
222222222222222
222222222222222
022222222222220
0000222222220000

```

```

18 MakeShadow 1Shadow
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000022222220000
002222222222220
022222222222220
002222222222200
0000022222220000
0000000000000000
0000000000000000

```

```

18 MakeShadow 2Shadow
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000022222220000
000222222222200
002222222222200

```

WELCOME TO CANADA!

- *Software Publishers
- *Peripheral Manufacturers
- *Hardware Developers

**Be Represented by Canadas Premier
Distributor of Amiga support products**

PHASE 4 DISTRIBUTORS INC.

**HEAD OFFICE: 7157 Fisher Road South East
(403) 252-0911 Calgary, Alberta Canada T2H 0W5**

CAVALRY-TORONTO-VANCOUVER-ST.JOHNS

**ATTENTION:CANADIAN DEALERS
CALL TOLL FREE 1-800-661-8358
FOR THE LATEST AMIGA/128 UPDATES
or (403)-258-0844 for our Dealer BBS**

```
000222222222000
000002222220000
000000000000000
000000000000000
000000000000000
```

```

18 Makeshadow 3shadow
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000022222222220000
000222222222220000
000022222222220000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000

```

[illegible]

0000000000000000

[illegible]

CREATE ShadowVectors

```
] 0Shadow 1Shadow 2Shadow 3Shadow 4Shadow 5Shadow [
```

```
: ChangeShadow    ( vector -- )    2* ShadowVectors + W@EXECUTE ;
```

```
: FreeBall    ( -- )
    Ball +asEvenSprite +ssNum W@ FreeSprite
    Ball +asOddSprite  +ssNum W@ FreeSprite ;
```

```
: FreeShadow    ( -- )    Shadow +ssNum W@ FreeSprite ;
```

```
: Consecutive?      ( n\nn -- )    - -1 = ;
```

```
: ?Balls ( f -- )
  Ball +asEvenSprite +ssNum W@
  Ball +asOddSprite +ssNum W@
  Consecutive? NOT DUP
  IF Freeball FreeShadow THEN
  ERROR" Unable to allocate sprites" ;
```

```
: GetShadow  ( -- )
  Shadow 7 GetSprite 7 = NOT DUP
  IF FreeShadow THEN
    ERROR" Unable to allocate sprites" ;
```

```

: GetBall    ( -- )
  GetShadow
  7 4 DO
    Ball +asEvenSprite I    GetSprite I =
    Ball +asOddSprite I 1+ GetSprite I 1+ = AND
    IF LEAVE ELSE FreeBall FreeShadow THEN
  2 +LOOP ?Balls ;

```

```
\ Under 1.1 Kickstart, moving the even sprite moves them
\ both, but according to reports that has changed on 1.2
```

```
: MoveBallSprite (x|y --)
  ViewAddress +vViewport @
  LOCALS| viewport y x |
    viewport Ball +asEvenSprite x y MoveSprite
    viewport Ball +asOddSprite x y MoveSprite ;
```

```
: MoveShadowSprite ( x\y -- )
  ViewAddress +vViewPort @ Shadow 2SWAP MoveSprite ;
```

```
\ Executing this definition will set up the colors for the ball.
\ It will also change one color of the mouse cursor.
```

```
: 19-31.Greys ( -- ) \ Only for registers 19 through 31
ViewAddress +vViewPort @ 32
16 3 DO
  1- 2DUP I I I SetRGB4
LOOP 2DROP ;
```

\ These values were derived from a combination of geometry
 \ and fudging them until they worked.

```

15500 CONSTANT Xviewport
13200 CONSTANT Yviewport
500 CONSTANT Zmin
24575 CONSTANT Zmax ( 4096 / will return a value 0-5 )
319 CONSTANT Xmin
38465 CONSTANT Xmax
1152 CONSTANT Ymin
11712 CONSTANT Ymax
19392 CONSTANT Xcenter
6400 CONSTANT Ycenter
64 CONSTANT Gravity
32 CONSTANT HalfGrav
128 CONSTANT TwoGrav
95 CONSTANT Spring
VARIABLE Zvel
VARIABLE Zpos
VARIABLE Xvel
VARIABLE Yvel

: Perspective ( coord\center\viewport -- new coord )
LOCALS| viewport center |
center -
viewport DUP Zpos @ + */
center + ;

: Ycart ( y -- y1 )
Ycenter Xviewport Perspective -6 SCALE ( 64 / ) ;

: Xcart ( x -- x1 )
Xcenter Xviewport Perspective -6 SCALE ;

: Zcart ( -- vector ) Zpos @ -12 SCALE ( 4096 / ) ;

: MoveBall ( x\y -- x\y )
2DUP LOCALS| y x |
x Xcart Ymax Ycart OVER y Ycart Zcart DUP
WaitTOF ChangeBall ChangeShadow
MoveBallSprite MoveShadowSprite ;

: ClipX ( x\y -- x1\y ) SWAP Xmax MIN Xmin MAX SWAP ;

: ClipY ( y -- y1 ) Ymax MIN Ymin MAX ;

: ClipZ ( -- ) Zpos @ Zmax MIN Zmin MAX Zpos ! ;

: ClipToWindow ( x\y -- x1\y1 ) ClipX ClipY ClipZ ;

: -Yveladjust ( y -- y )
Yvel @ DUP * OVER Ymin - TwoGrav * - SQRT NEGATE Yvel ! ;

: Yveladjust ( y -- y )
Yvel @ DUP * OVER Ymax - TwoGrav * - SQRT Yvel ! ;

: AdjustVelocity ( y -- y )
DUP Ymin < \ off the top of the screen
IF -Yveladjust
ELSE DUP Ymax > \ off the bottom
IF Yveladjust THEN
THEN ;

VARIABLE Yrem \ Storage for velocity remainders
VARIABLE Xrem
VARIABLE Zrem
VARIABLE FrictionCoef \ Friction parameters
999 CONSTANT Air \ 0.1% friction loss in the air
990 CONSTANT Surface \ 1.0% friction when rolling

: Friction ( addr of remainder\velocity -- velocity1 )
1000 * \ Scale up the velocity
OVER @ + \ add the last remainder
FrictionCoef @ 1000 */
1000 /MOD \ break out the new remainder
SWAP ROT ! ; \ and save it away

: NewY ( y -- y1 )
Yrem Yvel @ Friction DUP Gravity + Yvel !
HalfGrav + + AdjustVelocity ;

: NewX ( x\y -- x1\y )
SWAP Xrem Xvel @ Friction DUP Xvel ! + SWAP ;

: NewZ ( -- )
Zrem Zvel @ Friction DUP Zvel ! Zpos @ + Zpos !

```

Font-A-Size

With Earthbound Software

Now available! Fonts of all sorts and sizes for the Amiga! Including: Broadway, London, Bubble, N.Y., Celtic, Boston, Cairo, Math symbols, Future, Moscow and many more... Only

\$11.95 / Diskfull!

Write for complete current list.

Jan. '87:

Strategic Defense-

Head to head missile command type action, or team up to defend the planet. Can also be played alone. Smooth graphics and digitized sound!!!

Send DS/DD Disk & \$1 S & H for free EBS demo disk w/sample fonts!

Font-A-Size:

Scales any font to any point size. Hides in the background and scales fonts as you load them in!

Only \$14.95

Coming soon:

Taskmaster-
Have some extra RAM?
Manage your machine with our multi-system master.



**Earthbound
Software**

Suite #237
1005 E. 60th St.
Chicago, IL 60637
(312)-667-8048

```

: DoMove ( x\y -- x1\y1 )
NewZ NewX NewY ClipToWindow MoveBall ;

: Blip ; \ Just as soon as I figure out sound!

: Reflect ( addr -- ) DUP @ Spring 100 */ NEGATE SWAP ! ;

: Enough? ( addr -- f ) @ ABS Halfgrav < NOT ;

: Stopped? ( y -- y\f )
DUP Ymax - Xvel @ OR Yvel @ OR Zvel @ OR NOT ;

: Front/Back ( -- )
Zpos @ DUP Zmin = SWAP Zmax = OR
IF Zvel Enough?
IF Blip THEN Zvel Reflect
THEN ;

: Sides ( x\y -- x\y )
OVER DUP Xmin = SWAP Xmax = OR
IF Xvel Enough?
IF Blip THEN Xvel Reflect
THEN ;

: Top/Bottom ( y -- y )
DUP Ymin = OVER Ymax = OR
IF Yvel Enough?
IF Blip ELSE Surface FrictionCoef ! THEN
Yvel Reflect
THEN ;

: Bounce ( x\y -- x\y ) Front/Back Sides Top/Bottom ;

: DrawBackground ( -- )
GINIT rport 1 SetApen ( same color as border )
2 10 moveto 201 69 drawto
2 188 moveto 201 128 drawto

```

```

637 10 moveto 438 69 drawto
637 188 moveto 438 128 drawto
438 69 drawto 201 69 drawto
201 128 drawto 438 128 drawto ;

\ define a custom screen with 2 bit planes
struct NewScreen BounceScreen
  BounceScreen InitScreen \ copy default values
  2 BounceScreen +nsDepth W! \ # bit planes
  CUSTOMSCREEN BounceScreen +nsType W!
structend

\ A non-movable, non-sizable window
struct NewWindow BounceWindow
  BounceWindow InitWindow \ copies default values
  0 BounceWindow +nwLeftEdge W!
  8 BounceWindow +nwTopEdge W!
  640 BounceWindow +nwWidth W!
  190 BounceWindow +nwHeight W!
  WINDOWCLOSE ACTIVATE | BounceWindow +nwFlags !
  fCLOSEWINDOW MOUSEBUTTONS |
  BounceWindow +nwIDCMPFlags !
  CUSTOMSCREEN BounceWindow +nwType W!
structend

: CleanupBouncer ( -- ) \ when fCLOSEWINDOW detected
  FreeShadow FreeBall
  CurrentWindow @ CloseWindow
  CurrentScreen @ CloseScreen ginit ;

: goodbye ( -- )
  ?turnkey IF bye ELSE abort THEN ;

: BouncerEvents ( -- ) \ process IDCMP events
  GetEvent
  CASE
    fCLOSEWINDOW OF CleanupBouncer goodbye ENDOF
  ENDCASE ;

```

```

: InitVelocities ( -- )
  Air FrictionCoef !
  0 Xrem ! 0 Yrem ! 0 Zrem !
  4000 Xvel ! 2000 Yvel ! 2000 Zvel ! ;

: Initialize
  0" Animation of an Attached Sprite in Multi-Forth "
  BounceScreen +nsDefaultTitle !
  GetBall Xmax Ymax ( first X and Y )
  BounceScreen OpenScreen verifyscreen
  CurrentScreen @ BounceWindow +nwScreen !
  BounceWindow OpenWindow verifywindow
  DrawBackground 19-31.Greys ;

: Bouncer ( -- )
  Initialize
  BEGIN InitVelocities
    BEGIN
      BouncerEvents DoMove Bounce Stopped?
    UNTIL
  AGAIN ;

: tst0 ( x\y -- )
  initialize 2DROP
  BEGIN xmax 1+ xmin
    DO I xpos ! BouncerEvents MoveBall
    10 +LOOP
  xmin xmax
    DO I xpos ! BouncerEvents MoveBall
    -10 +LOOP
  AGAIN ;

: tst1 ( x\y -- )
  initialize 2DROP BEGIN BouncerEvents MoveBall AGAIN ;

```

•AC•

2.2 SOFTWARE

HOME MANAGEMENT SYSTEM

For Small Business & Home Management

This superior AMIGA software package features an integrated data base that allows for information sharing among the four interrelated modules contained within the system's single software disk. Modules include:

- Financial Management
- Personal Calendar
- Mailing List
- Telephone Directory

Comes with complete documentation including a TUTORIAL and REFERENCE MANUAL. On-line help included on diskette.

For information call (312) 786-1410, or write:
Olamic Systems Corporation, Dept. TPT
141 W. Jackson Blvd., Chicago, IL 60604

(Canadian dealers call Phase 4 Distributors toll free at 1-800-661-8358.)

68000 MACROS ON THE AMIGA

"It means that the resourceful programmer has a practically unlimited ability to add new, powerful commands to his or her assembly language programs. "

by Gerald Hull

If you look at your assembler disk, you will see that it says "Amiga MACRO Assembler." This macro capability is no small potatoes. It means that the resourceful programmer has a practically unlimited ability to add new, powerful commands to his or her assembly language programs. In what follows, we will learn how these macros work, concentrating in particular on ones associated with the Amiga. In the process, we will learn how to make them dance to our own peculiar tunes.

By itself, the word 'macro' simply represents the opposite of 'micro'. We use a MICROscope to look into an organic cell, but when we study MACROevolution, we look outward to the forces ruling its function.

In the computer culture, 'macro' is used more specifically as the abbreviation of 'macroinstruction'. As such, it refers to an entire sequence of instructions invoked with a single command.

You will find "macro" capabilities touted not only in assemblers, but in spreadsheets, word-processors, and high-level languages like C and Lisp.

I like to divide computer people into two categories: spreadsheet people and game people. I'm a game person. I know nothing about spreadsheets, and cannot be trusted to lecture on that subject.

We will, however, take a look at C macros. They are clearly modeled after their assembly language counterparts, and provide a good way of illustrating the features of "macroinstructions" in general.

C MACROS

You can do monstrous, clever things with C macros. But even a look at a relatively simple C macro will help to illustrate some of the important features shared by C and assembly language macros.

C macros are created with the '#define' preprocessor command. There are three different things you can do with #define commands in C.

First, you can simply turn on other preprocessor commands, as when '#define FOO' turns on later code segments bracketed by '#ifdef FOO' and '#endif'.

Second, you can use it to declare a constant, similar to the assembler programmer's use of the EQU command. You are saying that such-and-such a character string will represent this-or-that value throughout the program. For instance, the command '#define WINDOWMAX 200' causes the C preprocessor to replace every occurrence of 'WINDOWMAX' with '200'.

Usage number three of '#define' is the one that provides something equivalent to assembly language macros. In fact, it is the very same mechanism brought into play by the first two uses. They simply involve more limited applications. A mathematician would express this by saying that the first two uses are "degenerate instances" of the third, an expression I find quite delightful.

What makes this third kind of #define statement so powerful and so tricky is its use of variables. Here is a simple example from the file 'INCLUDE/CLIB/MACROS.H' provided with the Lattice and Manx Amiga C compilers:

```
#define MIN(a,b) ( (a) < (b) ? (a) : (b) )
```

What such a #define statement says to the C preprocessor is this: "As you go through the rest of the text, every time you encounter the string 'MIN' with two arguments separated by a comma and enclosed in parentheses, replace it with the substitution string '((a)<(b)?(a):(b))'. At the same time, substitute the first and second arguments for 'a' and 'b' throughout."

So if somewhere in your program there's the line

```
least = MIN(least, table[i]);
```

by the time it gets to the C compiler, the preprocessor will have transformed it into

```
least= ((least) < (table[i]) ? (least) :(table[i]) );
```

In effect, then, this C macro allows you to use a convenient abbreviation to represent a much more complicated operation.

The variables in this example are 'a' and 'b'. In your use of the MIN macro, you can substitute practically any numeric expression for them. However, some expressions are best avoided in macros, such as ones involving functions or the increment and decrement operators. This is because, of course,

the "expansion" of the macro may use such expressions two or more times, with unintended consequences.

It is important to remember that these macros are no more than abbreviations or recipes. By themselves, they generate no code. Only if you call or "invoke" them somewhere else in your program, do they generate any instructions. Indeed, they expand into a new, separate patch of code each time they are invoked.

68000 MACROS

A 68000 assembly language macro provides the very same kind of flexibility. Essentially, it consists of any sequence of commands which begins with a label followed by the word **MACRO**, and ends with the word **ENDM**.

Such "macro definitions" are stored up by the assembler as it passes through your program. Once in store, they can be invoked by the labels that introduced them. Every time the assembler encounters such a predefined macro label, it will "expand" your program by inserting the instruction sequence associated with it. Figures One and Two illustrate this deceptively simple mechanism.

The above C MIN macro can be rendered as an assembly language macro as follows:

```
MIN      macro

**      IN:      \1,\2    signed long ints
**      OUT:      \1      lesser if \1 != \2
**                      \2      unchanged

                cmp.l    \2,\1
                ble.b    \@
                move.l   \2,\1

\@
        endm
```

If you are typing this into your own program, make sure that the 'MIN' and the label '@' are left justified. That is, start them in the first column of text.

As with the C macro, this macro definition doesn't by itself generate any code. It is rather a "directive" to the assembler in its role as preprocessor, during its first pass through your code. It is saying: "Store up this sequence of instructions, treating backslashes followed by a number as slots to be filled later with character strings. And a backslash followed by '@' is to be replaced by '.nnn', where nnn is the number of macros expanded so far."

The slots created with '\1' and '\2' are equivalent to the variables 'a' and 'b' in the C example. And just like the high-level macro, the assembler will fill them with whatever you tell it to. So, in addition, our macro definition says: "Every time the string 'MIN' crops up later on, followed by character strings separated by commas, expand it by inserting the previously stored up sequence, with the first character string in the '\1' slot, the second in the '\2' slot, and so forth."

Consequently, when the assembler finds a line in your program which invokes this macro,

MIN d0,d1

it will insert the following "macro expansion":

```
**      IN:      d0,d1    signed long ints
**      OUT:      d0      lesser if d0 != d1
**                      d1      unchanged

                cmp.l    d1,d0
                ble.b    .007
                move.l   d1,d0

.007
```

Interestingly enough, as I have shown, the Amiga/Metacomco assembler will also expand the comments! This is because the assembler makes no distinction between code and comment during macro expansion. Your program is being treated simply as a piece of text to be processed according to certain rules.

The character string 'd0' goes into the slot created by '\1', and 'd1' goes into the '\2' slot. Finally, '@' has been replaced by '.007' throughout, since I am pretending that this is the seventh macro expansion performed so far.

MACROS VERSUS SUBROUTINES

Why use macros? At the very least, they provide an interesting alternative to subroutines. For comparison, let's look at a subroutine version of MIN:

```
**      MIN subroutine
**      in:      d0,d1    signed long ints
**      out:      d0      lesser if d0 != d1
**                      d1      unchanged
MIN      cmp.l    d1,d0
                ble.b    l$
                move.l   d1,d0
l$      rts
```

Such a subroutine would be typically invoked with a JSR:

```
<load up registers d0 and d1>
jsr      MIN
<extract the desired minimum from d0>
```

Unlike the macro version of MIN, this subroutine is not a recipe stored up by the assembler during the construction of your program. Instead, it is translated directly into code which takes up space regardless of whether it is ever executed. That is to say, a subroutine does not have to be called to exist in a program, as we sometimes discover to our considerable regret!

By contrast, any number of macros may exist in your program and never show up in the actual code it generates: the so-called "executable image." (Sounds terribly ruthless, doesn't it?) Because, just as with C macros, if you don't invoke them, they don't do anything. This is why, generally speaking, it does not matter if you have unnecessary Amiga library INCLUDE files in your programs, except to put the assembler into apparent coma. Those files consist largely of macro definitions.

But as well, a single macro may generate many pieces of code, one for each time it is invoked. And since every invocation of a macro expands into a distinct patch of code, they are inherently space-wasteful. Offsetting this, however, is the fact that they

are time-efficient. You don't have to JSR and RTS, or push and pop parameters, or maybe even save and restore registers, each time you invoke them. Subroutines, of course, while space-efficient because every call jumps to the same code, are complementary time-wasteful.

In our little MIN subroutine, we did not have to worry about pushing parameters on a stack. But we did have to make sure that d0 and d1 contained the values we wanted, which amounts to nearly the same thing. In practice, this would likely require additional MOVEs, since those registers would probably perform other functions elsewhere.

When using the MIN macro, however, we are free to use it on whatever registers or memory locations we please, so long as we don't fall afoul of the addressing modes for CMP and MOVE. So in that respect, macros boast an extra element of flexibility.

Despite these differences, macros and subroutines share a number of features. One, they provide a convenient means of breaking a program up into self-contained, single-function modules. Two, in virtue of that modularity, they can render programs much more readable and auto-documenting. And three, they can be gathered up into libraries of well-tested sequences which can be reused in other programs, perhaps by other programmers.

To sum up, in those instances where macros and subroutines are both plausible means of performing some function, subroutines are recommended when it is important to conserve space. However, if you're not worried that the multiple utilization of that function will exhaust your memory, macros can provide important advantages of speed and flexibility.

DIRECTIVES AND PSEUDO-OPS

So macros would be a pretty useful thing, even if all one could say on their behalf is that they provide an important alternative to subroutines.

But in fact, 68000 macros can do things mere subroutines cannot. For, as we have seen, macro definitions are classified as "assembler directives." As such, they have their impact during the very process of constructing your program, and not merely when it executes. There are many such directives, also called "pseudo-operations," or "pseudo-ops" for short.

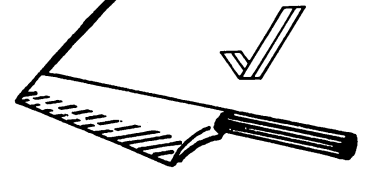
So there are two different kinds of commands you can use when you are writing 68000 code. First, there are the regular instructions, which tell the processor what operations to perform when your program is run: MOVE's, JMP's, ADD's, and so forth.

Second, there are these pseudo-ops or assembly directives, which have nothing to do with the execution of your program, but instead control the assembler's behavior during the process of generating machine code. Here we have commands like NOLIST, which turns off listing file production; DC.L, which allocates a longword of storage space; and indeed the MACRO and ENDM instructions which enclose a macro definition.

For non-interpreted languages like C and 68000 assembler, as contrasted with Basic, the process of getting code up and running can be divided into two distinct stages.



**WE'VE
SLASHED
PRICES
ON THE AMIGA!**



The West Coast's Largest Inventory of Commodore And Amiga Software And Hardware Products

CALL  (818) 366-5305 • 366-9120

10815 Zelzah Avenue Granada Hills, CA 91344

There is the program construction stage, during which the code gets translated into an executable image, and the program execution stage, during which that image makes the computer in fact do something. FigureThree illustrates this important distinction. The point is this: just as assembler commands can be concerned with either one or the other of these two stages, so can your assembler macros.

And during the constructive stage macros can provide forms of subtlety and power no subroutine, indeed no portion of an executing program, could possibly possess. They give you control over the program before it executes. Although we'll be looking at both "constructive" and "executive" macros (as you could distinguish them), I'm going to emphasize the former. It is here, I feel, that assembly language macros truly come into their own, and where the 68000 Amiga macros particularly shine.

By the way, to induce the assembler to provide you with listings of your 68000 programs, if it does not already do so, you want to add option '-l' ("dash L") to the ASSEM command. This is what I have in my assembler MAKE file:

```
:c/assem <file>.asm -o <file>.o -l <file>.lst + -l
:include -c W80000
```

This generates '.LST' files which show exactly how macro definitions and the other assembler directives have affected the assembled program. The 'W80000' has the effect of expanding the workspace available. You will need the larger space to accommodate your more ambitious assembly language efforts.

68000 AMIGA MACROS

We finish up with a look at some of the macros intimately involved with 68000 assembly language programming on the Amiga. Listing One begins with group of executive macros from EXEC/TYPES.I. The Amiga operating system consists largely of doubly linked lists of specialized structures. As you can imagine, it is important to manipulate and traverse those lists as quickly as possible. So instead of subroutines, these utilities are provided in the form of macros.

However, we will not attempt to peer into the details of what these particular macros are doing. Doubly linked lists can be very confusing, as anyone who has attempted to understand Chapter One of the ROM Kernal Manual will attest. (It is not your fault - there are a large number of flat-out errors in that part of the RKM. Perhaps a future article will address the subject.)

The next bunch of macros, beginning with CALLLIB, are all variously involved in invoking Amiga ROM Kernal modules. In order to make the operating system as flexible as possible, different functions are grouped into libraries. Every module in a particular library is accessed through a table of vectors determined through that library's base address. (Please note, however, that so far as memory is concerned, this "base" address comes at the end of the table, not the beginning; see below.)

If you are working in C, the only glimpse you are likely to get of this feature of the Amiga system architecture is the need to call OpenLibrary(), which attempts to load the library modules and associated vector table into memory, and returns the address of its base. At this point, all you need to do is make sure that the address isn't zero -- which would signal that the loading attempt failed -- and remember to use CloseLibrary() when you're done.

If you are working in an assembler, on the other hand, you will probably have to become more deeply involved in the process. You still have to call the assembly language equivalents of the OpenLibrary() and CloseLibrary() routines, unless you are linking your assembler into a C program and can lean on C library handling.

However, you will also have to call the various ROM Kernal functions via the "library vector offset" (LVO) protocol. This protocol requires, first, that you move the appropriate library base address into register A6. For example

```
move.l    GfxBase,a6
```

That particular register is necessary, unless you're a friend of the Flashing Guru, because the functions you call will feel free to use the contents of A6 to access their library siblings.

Now you prefix the name of the particular module you wish to call (say, 'Draw') with the string '_LVO', and JSR using "address register indirect with displacement" addressing:

```
JSR      _LVODraw(a6)
```

Figure Four illustrates this type of addressing on the Motorola 68000. In addition, you must have previously defined your offset vector -- '_LVODraw' in our example -- either through an equate (EQU) or as an external reference (XREF).

CALLLIB, LINKLIB, ETC.

All of this seems pretty messy, you're thinking, and of course you're right. So we are provided with some macros which, in various ways, smooth over all that mess. The first two, CALLLIB and LINKLIB, allow you to forget about addressing modes. Assuming that you have already opened the appropriate library (say, for graphics) and loaded up A6, you can simply say

```
CALLLIB   _LVODraw
```

Again, assuming the library is open, however A6 has been used for something else, you can still say

```
LINKLIB   _LVODraw,GfxBase
```

where GfxBase is a longword into which you've stored the base address. The macro will correctly set up A6 for you, and return it containing whatever it had before you called LINKLIB.

To protect the unwary programmer against confusion, these macros use the special symbol 'NARG' and the 'FAIL' directive to check that the proper number of parameters have been passed in. For example, the sequence

```
IFGT      NARG-1
FAIL
ENDC
```

will cause the assembler to generate an error message if somewhere CALLLIB has been called with more than one parameter: "Error 122: User 'FAIL' Statement."

It does so through a "conditional assembly" capability that will also be familiar to C programmers. Just as '#ifdef FOO . . . #endif' removes code from compilation, 'IFGT . . . ENDC' is just one way the 68000 programmer can exclude code from assembling. Here we are checking to see IF NARG-1 is Greater Than zero. NARG is a special symbol representing the number of parameters, separated by commas, contained in the macro's invocation.

However, this conditional assembly capability is a lot more flexible than the C version. Other possible conditions include IFEQ (IF Equal), IFNE (IF Not Equal), IFGE (IF Greater or Equal), IFLT (IF Less Than), IFD (IF Defined), IFC (IF strings are identical - Coequal?), and so forth.

But, back to the story! You can even forget about all the LVO business by using CALLSYS:

```
CALLSYS   Draw
```

This macro shows, as does LINKLIB, that there is nothing to prevent one macro from calling another, as long as this "nesting" is no more than ten deep.

Note that CALLSYS presupposes that A6 has been correctly set up. However, there's nothing to prevent you from defining a version that calls upon LINKLIB instead. And while you're at it, you could put in a conditional assembly to ensure that you don't send too few, as well as too many, parameters:

```
LINKSYS MACRO      * FuncOffset, LibBase
                IFLT      NARG-2
                FAIL      LINKSYS - too few args
                ENDC
                LINKLIB    _LVO\1,\2,\3
                                ENDM
```

To use this high-test version of LINKLIB, all you'd need to say is

```
LINKSYS Draw, GfxBase
```

Two other macros, XLIB and FUNCDEF, address the task of defining your LVO references for the assembler. In fact, they provide two different ways of doing this.

XLIB is the first way. You tell the assembler to regard your "_LVO" references as external, for example:

```
xlib      _LVODraw
```

This means that the linker needs to find these expressions defined elsewhere. You can provide this by including LIB/AMIGA.LIB as a library file in the linking process; as for instance:

```
:c/blink <file>.o to <file> library :lib/amiga.lib +
map <file>.map
```

By the way, this use of BLINK will also produce a '.MAP' file which will list all of the LVO references in any of the libraries you have used.

BLINK, for those of you who haven't yet acquired it, is a public domain replacement for ALINK released by John Toebes and The Software Distillery. It does everything that ALINK does and more, yet is anywhere from 2 1/2 to 6 times faster. You can find it on the AMICUS and Fred Fish disks, and most Amiga bulletin boards. I heartily endorse it!

The second way to define your LVO references is to use the FUNCDEF macro with a specially sequenced list of invocations. For instance, if your copy of the Amiga Macro Assembler is like mine, it contains a "read-me" which tells you to include EXEC/FUNCDEF.I prior to any inclusion of EXEC/EXEC_LIB.I, which contains the list in question. There is a similar macro and list in LIBRARIES/DOS_LIB.I.

As you can see, FUNCDEF contains a special variable, FUNC_CNT, which is initialized (SET) immediately following the macro definition. Each time FUNCDEF is called, it prefixes the ROM module name sent to it with '_LVO', and equates the result (using EQU) to the current value of FUNC_CNT. Then the latter is decremented by 6 in preparation for the next FUNCDEF call. Thus, as alluded to earlier, the LVO vectors are negative offsets to the library "base" address.

The only problem with FUNCDEF -- if your version of the assembler's INCLUDE files is the same as mine -- is that the value used to initialize it ($4*6 = -24$) is wrong. I have changed this to the correct value in the listing: $5*6 = -30$. However, just to be on the safe side, I never use FUNCDEF, preferring the (to me) more reassuring XREF route.

DISK DRIVE Expansion Box \$225.00.

Features:

- *Same size and color as Amiga™.
- *Mounts directly on top of the Amiga™.
- *Includes 40W Switching P/S,
- *Floppy Interface PCB, and
- *Amiga™ Interface Cable.

EB1000	Cabinet for one 3.5" & one 5.25" drive	\$225.00
EB1000-1	Cabinet with one 3.5" FDD	\$350.00
EB2000	Cabinet for two 3.5" Drives	\$225.00
EB2000-1	Cabinet with one 3.5" FDD	\$350.00
EB2000-2	Cabinet with two 3.5" FDD's	\$475.00
EB3000	Cabinet for two 5.25" Drives	\$225.00
IB200	Floppy Interface PCB	\$35.00

Ordering Information: Include \$3.00 for shipping and handling plus \$0.50 for each pound over 5 lbs. Shipped UPS ground. No COD's please. California residents add 6.5% sales tax.

Send check or money order to:

STACAR International

14755 Ventura Blvd., Suite 1-812
Sherman Oaks, California 91403
Telephone # (818) 904-1262

THE STRUCTURE MACROS

Next we look at a whole series of interrelated macros beginning with STRUCTURE. Unlike the list processing macros we began with, these are what I have called "constructive" macros. You don't use them because they are faster. You use them to do things no subroutine could do.

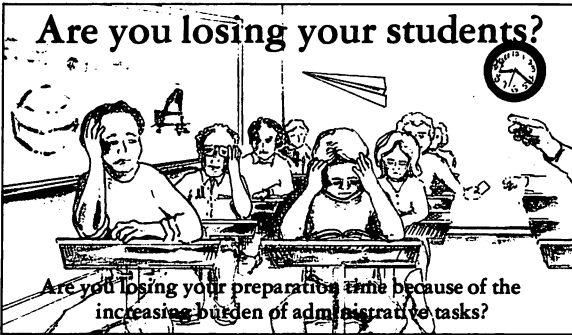
These particular macros enormously facilitate access to the special data structures which are at the heart of the Amiga operating system, linked together with the lists we touched on earlier. You can't do diddly on the Amiga without dealing with these structures, and the STRUCTURE macros make dealing with them as easy as in a high-level language.

For instance, in C the following defines the elements in the List data structure:

```
struct List
{
    struct Node *lh_Head;
    struct Node *lh_Tail;
    struct Node *lh_TailPred;
    UBYTE lh_Type;
    UBYTE lh_pad;
};
```

Such a definition makes it simple to deal with those elements. For example, you can say "List.lh_Tail = 0."

Are you losing your students?



Are you losing your preparation time because of the increasing burden of administrative tasks?

YOU HAVE CONTROL OF GRADE MANAGEMENT WITH GRADE MANAGER!

- C-64/128, PET — \$69.95; AMIGA — \$89.95
- Menu driven for immediate ease of use
- Teacher written and oriented documentation
- Prints out: Class Rosters, Attendance Rosters, Grade Book Sheets, Grade Lists with or without names and ID numbers, Individual Grade Sheets or Progress Reports, Midterm and Final Averages, Screen Dumps
- Disk Utilities for flexible file management
- Handles extra credit and incompletes
- Field searches through records
- HANDLES AN UNLIMITED NUMBER OF CLASSES
- 100 STUDENTS PER CLASS AND 42 GRADES PER STUDENT

Associated Computer Services • 1306 E. Sunshine • Springfield, Missouri 65804 (417) 887-7373

Yes, I want control of Grade Management. Send _____ copies of GRADE MANAGER C-64/128, PET — \$69.95; AMIGA — \$89.95, postpaid, to:

Name _____ Title _____

Home/School Address _____

City _____ State _____ Zip Code _____

Telephone () _____

Associated Computer Services • 1306 E. Sunshine • Springfield, Missouri 65804 • (417) 887-7373

We will ship C.O.D. for costs

In virtue of the STRUCTURE macros, dealing with List structures is just as transparent in assembler. Indeed, the definition seems almost identical:

```

STRUCTURE    LH, 0
  APTR       LH HEAD
  APTR       LH TAIL
  APTR       LH_TAILPRED
  UBYTE      LH_TYPE
  UBYTE      LH_pad
  LABEL      LH_SIZE

```

as does dealing with the elements: "CLR.L LH_TAIL(A0)".

How does all this work? Well, when you invoke the STRUCTURE macro itself, you zero out a special counter, SOFFSET ("structure offset"). Then, as you invoke each of the "data type" macros, for example APTR or UBYTE, they increment the counter by the number of bytes they require (4 and 1, respectively).

As a result, the parameter associated with each of them becomes EQUated with the size of the offset from the base address of the structure being defined. For example, LH_TAIL will equal 4. So if you have to deal with this element, you simply move the structure's base address into any handy register, and take advantage of indirect addressing with displacement. Again, take a look at Figure Four.

When you finish up with the LABEL macro, which itself adds nothing to SOFFSET, you end up with the amount of memory, in bytes, required to store the data structure in question. Hence, LH_SIZE will equal $4+4+4+1+1 = 14$. So if it is a component of some other structure, this size together with the STRUCT macro provides a means to augment the SOFFSET of the latter by the appropriate amount. The STRUCTURE macros, like any that are well-designed, make your programs self-documenting and much more readable, as well as easier to write.

BITDEF AND BEYOND

We finish up our survey of Amiga macros with a pair that facilitate the setting and clearing of signal bits: BITDEF and BITDEF0. There are two related pieces of information you want handy when dealing with such a signal: its mathematical value (say, 32), and its bit position (5th, starting with zero).

As the documentation in the listing makes clear, by invoking BITDEF with the appropriate information, you produce the desired pair of EQUates. This is an excellent example of macro documentation, by the way.

I will leave it as an exercise to the reader to analyze exactly how these macros produce the desired result. (Note, however, the way that '@' can be combined with a string -- '@BITDEF'. Voila!, 999 unique labels at your disposal.) To me they represent an amazing display of software ingenuity. However, I can't, for the life of me, figure out why such an intricate device is necessary.

For it seems possible to get the same result with a much simpler macro:

```

BITDEF      macro
  \1B_\2    equ      \3
  \1F_\2    equ      (1<<\3)
endm

```

However, the many bruises on my ego testify that things I can't figure out aren't always meaningless. Some circumstantial evidence suggests that this code might have originally been written for a different (and somewhat simpler) assembler. But I'm not betting on it.

There it is: the Amiga Macro Story, Part I (The Beginning). You can write the later episodes yourself. You likely have inferred, assuming you've made it all this way, that I regard assembly language macros pretty highly. I find utterly fascinating the ingenuity that shines through the best of them. My recommendation is that, after studying the Amiga macros real hard (even those in EXEC/TYPES.I), you go out and make up your own.

•AC•

The AMICUS Network

This past month, I turned off my Amiga and traveled to Los Angeles to the West Coast Commodore Association show.

By John Foust

In some ways, I am a disadvantaged Amiga writer. I want to bring you the latest Amiga news, and the latest-greatest public domain software - but I live in the Midwest, and the Amiga was born and raised in California.

Some Amiga enthusiasts live down the street from Commodore-Amiga. They know the nightspots where employees gather. They hear who might be out of a job, and where so-and-so is working now.

My sole link to the left coast is the cold glow of my monitor screen, and the red eyes of my modem's lights. This past month, I turned off my Amiga and traveled to Los Angeles to the West Coast Commodore Association show.



A conference at the WCCA

WCCA in LA

On September 20 and 21, the West Coast Commodore Association presented the Los Angeles Commodore Show at the Airport Hilton.

The WCCA show differs from many computer conventions. The exhibitors are free to sell their wares from the booth. There were a lot of open wallets in the crowd, and people carrying bags and bags of new Amiga software, Amiga magazines, Amiga t-shirts, Amiga bumperstickers, and Amiga socks. No kidding. Even Amiga sweat socks.

The show wasn't Amiga-specific, only Commodore-specific. About half the booths carried Amiga products. The Commodore 64 and 128 were in full force, but I didn't see any PETs behind a booth. I'm sure there was one, just to remind people of the good old days.

WCCA speakers

Aside from the exhibitor areas, the WCCA show also featured lectures from names well-known in the Commodore world, such as Jim Butterfield, of Compute! magazine and books, Richard Immers, author of several nitty-gritty Commodore 64 programming books, Len Lindsay of the COMAL user group, RJ Mical, former Commodore-Amiga programmer, and William Volk, of Aegis Development.

Aegis

Aegis Development's Bill Volk demonstrated Draw Plus, an upgrade to Aegis Draw, and evangelized about the Aegis user interface style. He showed demo screens from future products, including a game called "Defender of the Crown" (Designed by Master Designer Software as part of their Cinemaware™ and to be marketed by Mindscape.)

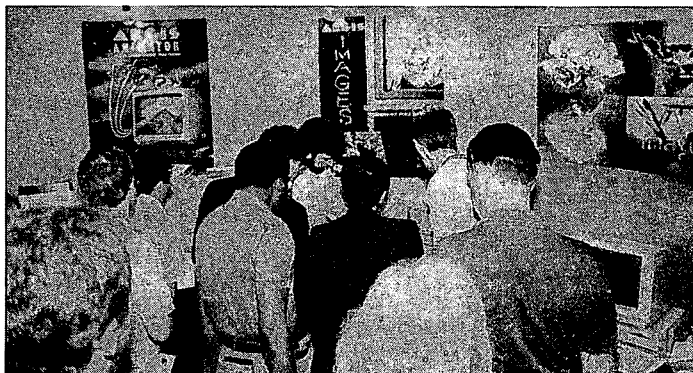
Artist Jim Sachs spent as many as several weeks on individual background pictures for soon-to-be-released Defender of the Crown. It is an interactive adventure game centered on the theme of the Robin Hood adventures.

Volk demonstrated several example sequences from the game. The video artistry was seamless. In one scene, Robin's gang is fighting a band of rogues in the courtyard of a castle. Your character leads the actions of the other characters. As your character backs away from the scene, through an open door, the viewpoint changes, and the fight continues up a staircase. There is even a love scene.

Areas of scenes are animated, apart from the actions of the characters. Flames flicker, and birds fly in the sky. Many scenes showed evidence of the Amiga's dual playfield mode, where video images overlay one another, and scroll independently. This technique greatly enhances a sense of motion and depth in Sachs' paintings.

The techniques of video art can be difficult, however. "It's hard to get something that looks like a horse using about six pixels," said Sachs.

RJ Mical is one of the programmers for the project. Aegis secured him because he knows the intimate details of the Amiga's low-level animation routines. Mical's wit was present in some of the demo screens. Mical formerly worked for Williams as a video game designer, and one of the screens looked like the opening of the Defender video game. Another was labeled "Defender of the Crow."



Aegis demonstrated Defender of the Crown

Volk also said they are working on a program that translates Animator sequences directly to a form that can be used directly by programmers. They also plan to release the details of a text-based script language that serves as the undercarriage of Animator scripts. This method would allow the creation and editing of an animation sequence using a everyday text editor.

RJ Mical

Intuition programmer RJ Mical held several Amiga-oriented lectures. Much of the time was spent fielding questions from the audience. Asked to compare programming languages, Mical said C is "a great language for programmers that pretend to be human beings," and that BCPL is "a language that will make your teeth hurt. Avoid it if you can."

Many questions were directed at his involvement with the software of the Sidecar. Commodore-Amiga techie Dale Luck was in the audience, and confirmed that the Sidecar had passed FCC clearance the week before the show. When asked if Microsoft Windows, an IBM windowing system, would run on the Sidecar, Mical said yes, and that "it's great to have Microsoft Windows in an Amiga window, especially when you make it real small, and push it to the back."

The ROM-based Amiga

Mical also confirmed Commodore-Amiga engineers have completed the ROM-based Amiga. When powered on, this Amiga would not need a Kickstart disk, only a Workbench disk. This change was made to lower the overall cost of the Amiga. The present system, with the Kickstart in RAM, adds an estimated \$60 in parts, which translates to a \$200 increase in the consumer cost of the Amiga, he said.

Commodore-Amiga programmers shoe-horned the new 1.2 operating system into ROM at the expense of some luxuries in the present Kickstart code. The names of the Commodore-Amiga programmers were present in previous Kickstart code images, but the names were removed in order to make more space in the ROMs. "It was that tight," according to Mical, explaining that only a few bytes were left free in the ROMs.

Does a ROM-based Amiga mean it can never use a new version of the operating system? According to Commodore engineers, the ROM-based Amigas will recognize a Kickstart disk, if it is inserted immediately after power-up. The machine will then disable the Kickstart-in-ROM, and use some of the machine's RAM space to hold the newer Kickstart code.

In other words, at the expense of using RAM space, a ROM-based Amiga will be able to use newer operating systems. There is always the potential of ROM upgrade kits, also. The yet-unannounced 'baby Amiga' is expected to be ROM-based. It is not known whether the present form of the Amiga will ever be ROM-based.

Byte-by-Byte

Several booths showed the Byte by Byte PAL expansion box. This comes in several flavors that support up to 8 megs of RAM, ST-506 and SCSI hard disks, from 20 to 40 megabytes storage, Zorro expansion boards and clock/calendar options.

Byte by Byte also showed the PAL Jr., a box with a 20 megabyte DMA hard disk, one megabyte of memory and a SCSI port, all for \$1,495. It auto-configures under AmigaDOS 1.2.



Jim Becker with Infominder and PAL

Jim Becker, of Terrapin Software, showed a version of his Infominder product in the Byte by Byte booth. The program linked the Amiga to a laser disc player. The twelve-inch laser disk contained more than 2400 images of paintings from the National Gallery of Art, and Infominder was loaded with an index of the paintings.

As you selected a leaf from the outline, the painting was displayed on the television next to the Amiga. You could select a list of all paintings by artists whose names began with 'S', and then select pictures of George Washington, and see Gilbert Stuart's paintings. This version of Infominder is called Infominder Plus, and will be available this fall.

C Ltd. aMega card

Ed Lippert and Breck Ricketts were the founders of Cardco. Perhaps you remember them as printer interface manufacturers for the Commodore 64. They now head C Ltd., and today their products enhance the Amiga.

"We came out here to sell boards. We brought 20 boards with us. We figured if we sold 10, we'd be doing great. By 3 o'clock, all twenty were gone. We took orders for another half-dozen, and called back to Wichita, to have them air-freight 25 more, and it looks like those will be gone by the end of the day.

Lippert said C Ltd. has been very careful to meet the specifications set down by the engineers at Commodore-Amiga. The one megabyte aMEGA card has a list price of \$549, and auto-configures under AmigaDOS 1.2.

The C Ltd. 'rumor sheet' forecast several products from their future line, such as a 6 slot expansion box and a typesetting package for the HP LaserJet laser printer, called LJ Typesetter. This software includes several hundred licensed fonts, and a simple dot-command driven composition system.

At press time, C Ltd. announced their 20 megabyte hard disk. It is non-DMA SCSI disk, with pass-through. The price is \$995.

Comspec

The Comspec memory expansion board was the first memory board available for the Amiga. The first model was available in November, and the present auto-config model came out in April. Comspec's Meyer Toole stressed that his product is different from other boards in the Amiga market. It uses OKI surface-mount chip modules, nine in all, for a total of 81 chips in a small area. Meyer Toole explained that these chips consume much less power than conventional chips, and spread less radio interference.

Unfortunately, OKI raised chip prices under the order of recent US Department of Commerce rulings. Toole estimated it would triple the cost of the chips, in the long run. Comspec showed one and two megabyte RAM boards, at \$749 and \$999, subject to change.

New Tek

New Tek, the Digi-View people, showed several new products, and improvements to the current video digitizing software. New Tek's Tim Jenison said "The new software supports consumer color cameras. You still have to use the filter. It has a new hold-and-modify mode that cleans up the colored speckles."

What sort of improvements did he make to enhance the color camera image? Jenison said "The problem with color cameras was that the color signal itself has a high frequency carrier in the video signal that causes an interference pattern, because it was of a very similar frequency to the sampling rate. We found a software technique to filter out that color signal, so it looks exactly like a black-and-white camera."

"The [new] palette features will let you take a Deluxe Paint picture, load the palette into Digi-View, and Digi-View will match the picture to that palette, so you can take that picture back and make a brush out of it, and put it into the original painting without messing up the palette.

"You can also use any arbitrary number of colors as opposed to thirty-two before - you can go down to eight colors or two colors. You can construct your own palette with the red, green and blue sliders.

"That's part of the special effects project we're doing for low-end video applications. Even broadcast stations have shown a lot of interest in this. Even if they have the effects to do this, people are still lined up to use them, and rent time on these machines.

"We're taking a still image, and then geometrically manipulating it, revolving it, wrapping it around, that sort of thing. These machines, like ADOs and DVEs, cost generally a minimum of \$100,000. We're not going to replace those machines, because they can work with a live video signal. We are working with a still frame.

"For a lot of things, that's enough. If you are producing a commercial, you have a still product shot or a screenful of text. This special effects package is planned for release in about three or four months. Most of the effects will require expansion RAM, because they work by calculating the frame one by one, and placing them in high memory, and then pulling them in real time.

"We also have the motorized filter wheel for extremely lazy people, so you don't have to move your arm and reach up to rotate the wheel by hand. Lazy people aside, a lot of people have their camera some distance from the computer, and it's a lot of work to run back and forth.

What are some of the interesting applications of the Digi-View? Jenison claimed "Most of them don't really want to talk about it, or if they do want to talk about it, they swear me to secrecy first, because they have these get-rich-quick schemes. There are a lot of those, a lot of vertical market niches that the Amiga makes possible, because of the low price, and the fact that you can display a color photograph on the screen.

"For example, a lot of medical uses have come up, such as transmitting medical images over the phone. The neatest ones are the point-of-sale terminals. For example, you put up a picture of a person, and you change their lipstick color, or put eyeglasses on them. There are so many applications that I never expected."



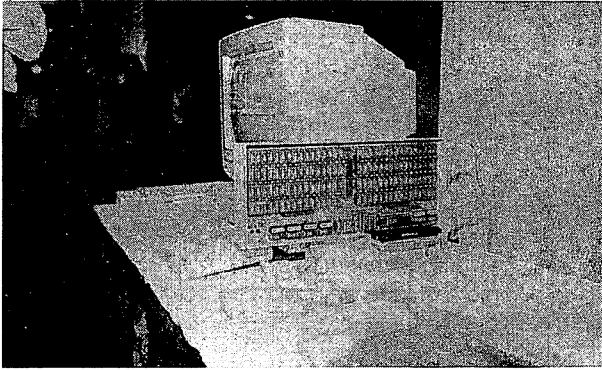
First Amiga Users Group (FAUG) booth

FAUG

Another popular booth was the First Amiga User Group, mainly due to the flashy demo software shown. This software included the SubLogic Flight Simulator, and Deluxe Music Construction Set.

FAUG is based in the Belmont area. Paul Montgomery, a founder of FAUG, and now an employee of Electronic Arts, said "We go to shows like this, whenever we can, when it will benefit the Amiga. We really don't make any money from them. We are here to promote the Amiga, because we think it's an awesome machine, and deserves to do great."

The FAUG meetings are well-attended. "We have between 250 and 500 people at our meetings. We've had Trip Hawkins speak, and Jay Miner did a meeting. Generally, though, about 300 people come to an average meeting. Jay Miner was our very first paid member... He comes to every meeting. He's there to talk to people, and answer questions."



RS Data System's eight megabyte RAM board

RS Data Systems

RS Data Systems makes an unusual RAM expansion for the Amiga. It is roughly L-shaped, and protrudes several inches in front of the right edge of the Amiga. The Amiga on display had the external drive tucked between the monitor and the RAM expansion.

According to Roy Eubanks, "It's the most expandable product here. It's the only one that goes to eight megs. It's more versatile for people who have access to their own RAM chips. We can give them a bare board, and they can upgrade to four megs, and then they can buy an additional board, and upgrade to eight megs."

"Admittedly, it is large, but for maximum expandability, you just can't get around that. Either you are going to grow outward, or make it one unit. If you start trying to stack them, then you run into problems with power. If they are cascaded on to each other, they reach a limit."

RS Data Systems sold several eight megabyte RAM boards at the show, at \$1675 each. The board is not auto-configure, but an optional auto-configure daughterboard is available. The two megabyte board lists for \$950.

Copperstate

Next to a stack of Amiga sweatshirts, Copperstate showed QuickNibble, a copy program that will both copy and de-protect the all latest Amiga software.

Somehow, with a straight face, a Copperstate employee explained that QuickNibble is "for archival purposes only", and then started the program.

After a title screen that explained the legality of making an archive copy, the program launched into a digital recording of the first seven notes of the Disneyland theme of "The Pirates of the Caribbean", which goes "yo, ho, yo, ho". He then explained that eight notes would have violated ASCAP's copyright regulations.

According to Dave Devenport, a programmer of QuickNibble, "the most common protection I've seen on the Amiga is what I call a 'sync track', where they use a non-standard index sync, and write an extra-long track. The Amiga can read twice the density that it can write. They try to read the sync track to see that it is an original disk."

Devenport explained the "Pirates" theme was all in fun. "Everyone understands the opening screen," he said.

Manx Software

Jim Goodnow, the author of the Manx Aztec C compiler, explained recent changes to the program development system. The code generator has been improved, and four floating point formats are available.

The linker now supports scatter loading. It also supports segmentation, a form of overlays. It is dynamic, but not the same as the tree-oriented overlays in 'alink'. The linker will also link the Metacomco object format. Although the new Aztec linker can use this format, register use conflicts still prevent the mixing of Lattice and Aztec code.

"We still have our own [object module] format, because it is much faster. I have much more respect for 'alink' now. It's not 'alink's fault, it's the object format's fault. It's just terrible." When using the 'alink' format, "doing a straightforward 'amiga.lib' link with my linker is twice as slow, as compared to 'alink'."

Goodnow's new debugger supports multiple tasks, multiple segments and scatterloading. It can trace function calls, indenting diagnostics and showing return values.

Goodnow said a source-level debugger is shipping now for his IBM PC compiler. He is using his Amiga to develop the code, and periodically uploads the code to his IBM PC to use the source-level debugger. An Amiga version of this debugger will be offered as an update in the future.

Developer area

The WCCA officials set aside part of the floor for developers, at a reduced cost, provided they didn't sell products. Jenday Software showed its Conversation with a Computer program, all 2000 lines of Amiga Basic. Zen Software showed a Amiga system monitor program that peeked around in the system task and device lists.

Spring WCCA in SF

Talking with the WCCA show organizers, I learned that the majority of the show's attendees heard about the show through Amazing Computing. This is incredible, since the only mention of the show was a short description at the end of my last column.

The next WCCA show will be in February 20 through 22 in San Francisco, at Brooks Hall. WCCA officials expect a larger crowd.

CES and COMDEX

Commodore will not be present at the COMDEX Fall convention in Las Vegas.

Commodore might be present at the January Consumer Electronics Show in Las Vegas. According to a source close to Commodore, Commodore has reassessed their marketing plans, and wants to present the Amiga and their other computers as consumer products.

World of Commodore

December 4 to 7 marks the World of Commodore show in Mississauga, Ontario, just outside Toronto, at the Toronto International Centre.

Like the WCCA show, it is a 'for sale' show. The two shows differ in size; the World of Commodore is huge, the largest Commodore show in North America.

The organizers of the WOC presented a video tape demonstrating the magnitude and popularity of last year's show. Last year, over 32,000 people visited the three-day show.

Since the tape was aimed at potential exhibitors, it explained the magnitude of sales one could expect. They set up temporary warehouse space on the show floor, so vendors can keep a smooth supply of goods behind the counter.

I hope to be there. Either way, Amazing Computing will be represented by Tim Grantham, former Amiga columnist for TPUG magazine. Grantham has a deep understanding of the Canadian Commodore community. (TPUG stands for the Toronto PET User Group, the largest Commodore user group in the world.)

The Amiga Zone

The People Link Amiga group was formerly a sub-section of the Commodore Club. In mid-October, it split off to form a new club, the Amiga Zone. In a matter of days, it grew to be the largest club on People Link. The Sunday night conference attendance surpassed all previous records. One Sunday, over sixty Amiga enthusiasts were online at one time.

Amazing Computing has a section in the Amiga Zone, as it did in the old club. When the new club formed, I was given a chairman position in the club. I don't think there is a conflict of interest in this, I'm not paid for my work there, but I do get free access to People Link for my work in the Zone.

Consider this balanced by the recent co-SYSOP appointment of Amazing Computing music editor Richard Rae. Rae will be assisting the SYSOPs on Compuserve's Amiga Forum. He gets the same treatment there as I do on People Link.

Our presence on these networks means you, the Amiga user, has even more contact with Amazing Computing. Our presence on the networks means better articles for you, since we are introduced to people doing new and interesting things with the Amiga. We can secure resident experts for review of products, and solicit general opinions and bug reports from Amiga users.

Telecom issue

Amazing Computing volume 2, number 1 will be the Telecommunications issue. It will carry reviews of the latest Amiga telecommunications products, and describe the national and local Amiga computer groups you can join with your modem. It will include the announcement of a special-purpose AMICUS disk telecom AMICUS disk, with a collection of the best telecommunication utilities.

Both commercial network computer systems and local, private bulletin board systems will be described. This will include instructions for registering an account on each service, a short guide to getting to the Amiga group on that network, and a synopsis of what you might find there.

ICUG's Source Amiga group

Deepak Midha and Larry Phillips are the primary organizers of the Independent Computer User Group, ICUG, a user service organization for Commodore computers at present, but for all computer users in the future. They have a number of exciting ventures planned for the future, aside from supporting all computer types.

They have an Amiga club on the Source. The Source uses a system called Participate to organize the topics of conversation within the ICUG area. I found Participate somewhat imposing at first. However, with a little practice, it is easy to use. The ICUG area also has a special area set aside for developers, called C.O.M.B.I.N.E..

To sign up for the Source, call 1-800-336-3366. If you mention this column, and give this waiver number, #6450110, you won't pay the regular Source signup fee, set at \$49.95.

The Well and Usenet

Perhaps you've seen messages on public domain disks with strange, unreadable sequences of nonsense words, separated by exclamation marks. This text was a message that travelled on Usenet, the world-wide Unix user network.

Usenet is one of the best-kept secrets of the network community. Imagine a world-wide computer mail system that gives door-to-door delivery, sometimes in minutes. Its participants are mostly university and Defense Department computer programmers, among the cream of the crop.

Because it's free to most users, people participate without reservation, and spend a lot of time on the system. It is my impression that many Usenet members spend an hour or two a day reading and posting messages.

If you weren't a college student or working for a contracted company, you had little chance of getting on Usenet. That has changed, with a system called the Well.

The Well is a Unix system open to the public, on a subscriber basis, much like People Link or Compuserve. The Well now has Telenet access. With Telenet's buyout of Uninet, more cities will have access to the Telenet and the Well. The rates are low, around \$3 an hour if you are in the Sausalito area. Add standard Telenet charges if you aren't.

If you aren't familiar with Unix, you can still use the Well, since the default user interface is command- and menu-driven, and help is available at any prompt. If you are a Unix user, you can drop out of the menu shell, and onto the bare metal of the '\$' prompt, so to speak.

How do you signup for the Well? Call your local Telenet node number. If you don't know it, call 1-800-TEL-ENET, and ask them. With your modem, dial this number, hit RETURN twice a moment after the modem detects the carrier tone. Hit RETURN again at the 'TERMINAL=' prompt. You will then see an at-sign prompt, '@'.

Enter 'C WELL', and when it asks for your 'login:', enter 'newuser'. You will be guided through a series of questions. Have your credit card number handy.

The signup procedure will ask who told you about the Well. If you answer 'jfoust', I will get a few hours' time added to my tab. I'd appreciate it. This obligates me to help you on the system, once you get there. To send me a message, just type 'mail jfoust' at most prompts on the Well. Enter 'g amiga' to join the Amiga group there.

AMICUS 14

The newest AMICUS disk is now available. It has several programs from past issues of Amazing Computing, including Daniel Kary's index to Amiga C programming structures, Mike Swinger's Amiga Basic program to convert small IFF brushes to BOBs, and Tim Jones' .bmap reader, including all the latest .bmap files.

There is also an AmigaBasic example of using autorequesters. Other programs include 'crlf', a filter to add or remove carriage returns to line feeds in documents, and 'queryWB', a program to get a yes/no response from the user during a startup sequence, and set an exit code.

'DosHelper' is a program much in the style of Kary's structure reference. It presents an Intuition menus with AmigaDOS commands. Selecting a command gives a screenful of tips on using that command.

There is a program to convert Commodore PET ASCII files to normal ASCII. If you have text files on your Commodore 64, you can use this program to convert them to text files on the Amiga.

Scientific American readers must like the Amiga. Yet another "Computer Recreations" column program has been converted to the Amiga. The September column features a program called C Squared. It generates screens of interesting patterns. The C source and executable are present, along with documentation.

'dpdecode' is a program to decrypt Deluxe Paint. It turns out that the 'dpaint.dat' file in the 'c' directory is the executable itself, and the 'dpaint' program is only a loader. An incredibly patient and wise programmer uncovered the encryption scheme, and wrote a program to convert your copy-protected Deluxe Paint into a non-copy-protected version.

'vc' is a visual calculator, in the style of VisiCalc. This is a simple spreadsheet. It doesn't use the mouse, only cryptic key commands. It is a port of the public domain Unix program of the same name.

'view' lets you look through a text file, using a window and a side scroll gadget.

There are four Oing-type programs here. If you haven't seen it, the 'oing' program bounces nine boing-style balls on the screen at once. Three other versions have appeared. One uses sound, so the original boing demo is more faithfully reproduced. One is like an air-hockey game, another is a chase-tag game.

There are three clocks on this disk. These are variations on a theme. They present themselves on the menu bar of the current Intuition window, and update the time periodically.

The text files include a tipsheet for Deluxe Paint, to make brushes from variable-shaped areas, an article on long-persistence monitors, and a list of suggested methods for icon user interfaces, from Commodore-Amiga.

New Fred Fish disks

At press time, as usual, the list of the latest Fred Fish disks arrived. They are listed in the public domain catalog in this issue. The next AMICUS Network column will discuss their contents.

Next issue

Next month, I will travel to the Amiga developer conference in Monterey, California, followed by COMDEX Fall in Las Vegas.

Every developer at the conference must sign a non-disclosure agreement. Presumably, Commodore will tell the developers about yet-unannounced products. I've promised to sign one, so my lips and typing fingers will be closed to expressing some thoughts in the future.

In some ways, this will be an advantage to the magazine. We'll be in on a few more secrets, and this buys more time to research, and get the latest Amiga news to you quicker than before. We won't be surprised when the 'baby Amiga' is launched, or when the Amiga 2500 is revealed.

Come to think of it, I don't think I'll be surprised about the announcements at the developer conference. The rumor mills will always grind, and I'm sure the announcements will be rumor-grist the night after the day's conferences, when people get on the telephone, or go back to their portable computers in the hotel rooms.

On top of that, COMDEX takes place the week after the conference, and many developers will travel to Vegas afterwards. As soon as someone says "What's new?," I'm sure some of the secrets will leak out.

Addresses:

West Coast Commodore Association

P.O. Box 210638
San Francisco, California 94121
(415) 982-1040

World of Commodore

The Hunter Group
204 Richmond Street West
Suite 410
Toronto, Ontario, M5V 1V6
(416) 595-5906

The Source

P.O. Box 1305
McLean, Virginia 22102
Customer info 1-800-336-3330
Signup 1-800-336-3366

People Link

3215 N. Frontage Road, Suite 1505
Arlington Heights, Illinois 60004
Customer info 1-800-524-0100
Signup (modem) 1-800-826-8855

TDI Modula-2 Amiga Compiler

*"Modula-2 is a new and powerful language,
and is a perfect match for the Amiga."*

Reviewed by
Steve Faiwiszewski

'The TDI Modula-2/Amiga is a state of the art high level language, simple enough for beginners to pick up easily, and powerful enough for serious programmers to write large complicated programs.' So claims the manual for this compiler. Before discussing the TDI package I must say a few words about Modula-2.

Why Modula-2?

Modula-2 is a general purpose language created by Niklaus Wirth, the author of Pascal. Wirth designed the language primarily for writing systems software. As such, the language is quite similar to Pascal but has the following advantages:

1. The language's syntax has been cleaned up, improving readability and efficiency.
2. Modula allows low-level programming without having to resort to assembly language, thus allowing the programmer to take advantage of unique system features.
3. Most importantly, Modula-2 introduces the concept of the "module", a programming technique which facilitates development of large systems and of multi-programmer projects.

Here are a few of the features that make Modula better than Pascal:

A. Open arrays: One of the annoying things about Pascal is that you can only pass a fixed size array as a parameter to a procedure. For example, if you have the following piece of code:

```
TYPE
  alfa40 = ARRAY [1..40] OF CHAR;
  alfa80 = ARRAY [1..80] OF CHAR;

VAR a : alfa40;
    b : alfa80;
PROCEDURE Print(one : alfa40);
BEGIN
END;
```

you can call Print only if the variable you pass it is of type alfa40. If you need to pass to Print a variable of type alfa80 you're out of luck; the only thing to do is to declare another procedure which accepts an alfa80 variable.

This makes it difficult (if not impossible) to write general routines in Pascal. Modula-2 addresses this problem by providing a means of

the open array. This simply means that you can pass any size array to a procedure, such as

```
PROCEDURE Print(VAR one: ARRAY OF CHAR);
BEGIN
END Print;
```

Now Print will accept any size array. You can find out the size of the passed array by using the Modula-2 function HIGH.

B. Type transfer. Like Pascal, Modula-2 is a strongly typed language. That means that, unlike C, you can assign the value of one variable to another only if both are of the same type. However, once in a while it is very desirable to eliminate type checking. Modula lets you do so in an orderly and controlled manner, unlike the haphazard way type casting is done in C.

In Modula-2 every type name can be used as a type transfer function. To change one type to another you simply use the type name as a function. For example to assign a value from a variable declared as WORD to a variable declared as INTEGER you would have:

```
VAR
  w : WORD;
  i : INTEGER;
BEGIN
  i := INTEGER(w);
END;
```

No conversion is done on the data; it is simply treated as the new type. Therefore type transfer only works between types of the same size (i.e. both types WORD and INTEGER take up 2 bytes, so you can transfer one to the other).

C. Short-circuiting of boolean expressions. In Pascal all parts of a boolean expression (such as an IF or WHILE statement) are evaluated. This could some times lead to trouble. Examine the following piece of code.

```
VAR
  i : INTEGER;
  table : ARRAY[1..Max] OF INTEGER;
BEGIN
  i := 0;
  WHILE (i <= Max) AND (table[i] <> 0) DO BEGIN
    .
    i := i + 1;
  END;
END;
```


You might notice that at one point the value of 'i' will exceed Max and then the expression '(table[i] <> 0)' will cause an "index out of bounds" error. Modula-2 corrects this problem by stopping the evaluation of the expression as soon as the value is determined.

In the above example, when 'i' exceeds 'Max' the expression '(i <= Max)' is false, and therefore Modula won't continue to evaluate the next part of the expression, and will never encounter the "index out of bounds" error.

D. Low level access. Unlike Pascal, which isolated the programmer from the computer, Modula-2 allows you to "get to the guts of the machine". There are features in the language to let you access the computer memory and hardware through bits, bytes, words and pointers (addresses).

The SYSTEM module (more about modules later on) defines such types as WORD and ADDRESS. WORD is a type whose size is equal to the word size of the machine (Actually WORD is defined as 16 bits in the TDI implementation, while LONGWORD is defined as 32 bits). ADDRESS is defined as POINTER TO WORD.

Some of the procedures usually defined in SYSTEM are: SIZE which returns the size of a given variable, TSIZE which returns the size of a given type, and ADR which returns the address of a given variable in memory.

Separate Compilation and Modules

By far the most significant change in Modula-2 is the ability to create and compile individual pieces of code separate from each other.

C also supports the idea of separately compiled modules, but it doesn't define any control over this feature. In C you declare in one module a function which accepts an integer, but from another module you may call it and pass it an array, and neither the compiler nor the linker will complain!

This will never happen in Modula, as it does type checking across modules. If you try to pass an array to a procedure which is defined in another module to accept an integer, the compiler will flag that as an error. Modula-2 also has version control. That means that if you changed something in the way a procedure is defined in one module, you will be forced to recompile any other module which might make use of the modified module. This feature guarantees that you will always be using the most updated code, and might save you hours of hair pulling.

The use of separately compiled modules is accomplished through the MODULE construct. There are 3 types of modules: main modules, definition modules, and implementation modules.

The main MODULE is equivalent to the Pascal PROGRAM, and contains the usual type, variable and procedure declaration in addition to the main line of code.

A DEFINITION MODULE contains declarations of types, variables, and procedures, but it has no code. The definition module serves to define an interface to routines that are found in a corresponding IMPLEMENTATION MODULE. All items defined in a definition module (constants, types, variables and procedures)

are "visible" to other modules, meaning that other modules "know" that these items exist, and might refer to them.

An IMPLEMENTATION MODULE contains the code of all the procedures defined in the definition modules, plus all other types, variables and procedures that are to be used only within the implementation module and are not visible to other modules. Additionally an implementation module might contain some initialization code that is to be executed once, before the main line of the program begins executing.

There is an IMPLEMENTATION MODULE for each DEFINITION MODULE. Any module that refers to an item which is declared in another module, must contain a statement which instructs the compiler to look the item up in the module it is declared in. This statement is known as the IMPORT statement.

For example suppose we have a module called DateAndTime which contains various procedures concerning dates and times. The definition module might look like this:

```
DEFINITION MODULE DateAndTime;

PROCEDURE DayOfWeek(n: CARDINAL; VAR name: ARRAY OF CHAR);

END DateAndTime.
```

And the implementation module might look like this:

```
IMPLEMENTATION MODULE DateAndTime;
  CONST MaxDayNameLength = 9;
  TYPE DayString = ARRAY[0..MaxDayNameLength] OF CHAR;
  VAR Week : ARRAY[1..7] OF DayString;

  PROCEDURE DayOfWeek(n: CARDINAL; VAR name: ARRAY OF CHAR);
  (* This will return the name of the n'th day of the week *)
  VAR
    i : CARDINAL;
  BEGIN
    i := 0;
    WHILE (i < HIGH(name)) AND (i <= MaxDayNameLength) DO
      name[i] := Week[n][i];
    INC(i)
  END;
END DayOfWeek;

BEGIN
  Week[1] := 'Sunday'; Week[2] := 'Monday';
  Week[3] := 'Tuesday'; Week[4] := 'Wednesday';
  Week[5] := 'Thursday'; Week[6] := 'Friday';
  Week[7] := 'Saturday';
END DateAndTime.
```

A module which makes use of the DayOfWeek procedure may look like this:

```
MODULE Test;
  FROM DateAndTime IMPORT DayOfWeek;
  VAR name : ARRAY[1..10] OF CHAR;
  BEGIN
    DayOfWeek(5, name); (* his should return 'Thursday' in name *)
  END Test.
```

Notice that the variable 'Week' is visible only within this IMPLEMENTATION module, and no other module can refer to it, since it is not declared in the DEFINITION module. This ability to "hide" information within a module is known as data hiding (or information hiding) and is one of the more important benefits of using modules. Data hiding is desirable when writing large programs, or when more than one programmer is involved.

In the example above, the person who wrote MODULE Test does not have to concern himself how DayOfWeek works. Data hiding also allows you to have what is known in C as static variables: variables which are only visible to certain procedures, but which maintain their value even when those procedures are not running. Notice that there is some initialization code which sets the Week array to the names of the days. This code will run before the main line of the program starts running.

The manner in which you would compile the above example is as follows: First you compile the definition file (using the TDI compiler, it would be called DateAndTime.Def). The compiler will produce a special symbol file (called DateAndTime.Sym). This symbol file is used by the compiler whenever you compile the implementation module, or any module which imports from DateAndTime. Next you would compile the implementation module (called DateAndTime.Mod), and the compiler would produce DateAndTime.Lnk.

Once you compile all your Test.Mod (and any other modules it might use) you are ready to link them. The linker links all the necessary .Lnk files into one executable file (called Test). Note that once you compile a definition file, you don't have to recompile it again even when you change some code in the implementation module (as long as you don't change the way a procedure is declared).

Advanced Features

Modula-2 supports some advanced features, such as coroutines and interrupt handling. These features are used to implement multi-tasking and other things related to operating systems. These features are beyond the scope of this article, but suffice it to say that whole operating systems HAVE been written in Modula-2.

As you can see, Modula-2 has combined the strict type checking and control of Pascal with the flexibility and power of C, yet it managed to avoid the disadvantages of the two older languages while introducing some new concepts. Modula-2 is a state-of-the-art language, and what is more befitting than to use it on a state-of-the-art machine such as the Amiga?

The TDI Modula-2 Amiga Package

TDI Software, Inc. markets a Modula-2 compiler package for the Amiga, as well as for the Atari ST and the Pinnacle.

The TDI Amiga package consists of a multi-pass compiler, a linker and a few other utilities, and it's a faithful implementation of the language, including such features as coroutines and interrupt handling. Most importantly the compiler allows access to all the ROM Kernel routines.

System Requirements

The minimum system requirements to run the compiler is an Amiga with 512K and one disk drive. However, I would very strongly recommend that you get two drives if you are going to use the package for anything more than just playing with it.

What You Get

Release 2.00a of the TDI package comes in three flavors:

1. The Regular version sells for \$89.95 and comes with one disk which contains the compiler, linker, error lister utility, a large number of support files (.Lnk & .Sym files), and a spiral bound manual.
2. The Developer version sells for \$149.95 and comes with everything the Regular version has plus another disk which contains the source for all the definition files (for all the .Sym files), some demo programs, additional modules that handle IFF and ILBM, and a bunch of utilities (a decoder of symbol files, a disassembler for link and load files, a cross-reference utility, and a kermit file transfer utility). This version also contains the source of some of the support modules (such as the InOut and Streams modules).
3. The Commercial version contains all of the above plus a third disk which has the source to all the support modules.

The disks are not bootable (that is they don't contain any of the WorkBench stuff on them), so before you can use the compiler you have to set up a couple of work disks. The package comes with installation instructions for a system with one or two disk drives, but I came up with my own setup, which I found to be more useful. I have a Workbench disk in df0: which contains the compiler and editor in the c: directory as well as my favorite editor (MicroEmacs), and any other program I might use. In df1: I have a disk which contains all the .Lnk and .Sym files (in a directory called M2) as well as any source code I'm working on.

The M2 directory is huge, so once in awhile I have to make room on this work disk by moving finished or old code to another disk. I assign T: to RAM: and this way the compiler uses the RAM disk for its work space. When it's time to link, I copy the main .Lnk file to RAM and link from there. This cuts down on head seeks and speeds up the linkage time while reducing the wear and tear on the drive head.

Once I get more memory, I might change this setup and put more things in RAM, but with only 512K, you can't do much better than this setup. The manual gives instructions how to use the package on a one-drive system, but due to the size of the compiler and the M2 directory, I found that to be a real inconvenience.

Running it

Once you create the definition and implementation modules (as well as the main module) for your program you invoke the compiler by typing 'modula <filename>' where <filename> is the name of the module to compile.

As an example let's assume that you have created a module called Test. So you have two files: Test.Def (the definition module) and Test.Mod (the implementation module). First you must compile the definition module by typing 'modula test.def'. If there are no errors, the compiler will create the symbol file Test.Sym in the directory to which you are presently connected.

If any errors were encountered the compiler will produce an error file called Test.erd. You can list the errors using the M2ERROR utility by typing 'm2error test.def'. M2Error will display the offending line in the module and the type of error encountered in that line. I usually run M2Error as one task, and my editor as another so I can correct the errors as they are displayed in M2Error's window.

Care must be taken, though, not to save the changes before M2Error completes running, as it keeps the source file open during its execution, and modifying the source file on disk can confuse M2Error and you might encounter the Guru!

Once you're done with compiling the definition module you would compile the implementation module by typing 'modula test'. Note that if you don't specify an extension to the file name, the compiler defaults to '.Mod'.

If any errors are encountered the compiler will generate Test.Erm, otherwise it will produce Test.Lnk. Whenever the module being compiled imports something from another module the compiler has to read the symbol (.Sym) file for that other module. The symbol file is first searched for in the current directory, and if it wasn't found then the search continues in the M2: directory.

The compiler can optionally prompt you for the location of the .Sym file, and it can also produce a listing file for the module being compiled. The listing file contains the source lines of the module, and addresses of statements. This information might be useful during debugging.

After you compile all the code into .Lnk files you link them into an executable program by invoking LINK. As with the compiler, Link first searches for the appropriate .Lnk file in the current directory, and if it can't find the file, then it searches the M2: directory.

The linker can also optionally prompt you for the location of the .Lnk files. LINK is also capable of producing a listing file (known as the Map file) of the program which contains addresses of the various modules. Again, this information might be useful during debugging.

The linker supplied with Release 2.00a of the package also has the option to optimize the generated code, which can shrink the size of the resulting executable file significantly. Using the Optimize option will increase linkage time a bit, but you can get as much as 80% reduction in file size!

Standard Features

The TDI compiler implements most of Modula-2's standard features. Standard types INTEGER and CARDINAL are represented by 16-bit words. The LONGCARD and LONGINT types are implemented as 32-bit longwords, the standard type BITSET is 16-bit long, and the standard type CHAR is one byte. All pointer types are 32-bit longwords, as the 68000 uses longwords for addresses.

The SYSTEM module defines such types as BYTE (8 bits), WORD (16 bits), LONGWORD and ADDRESS (32 bits). The ADDRESS type is compatible with all pointer types. SYSTEM also defines the SIZE, TSIZE, and ADR functions.

The SYSTEM modules also defines procedures such as NEWPROCESS, IOTRANSFER, TRANSFER, and LISTEN. These procedures are used for Modula-2 multi-tasking (not to be confused with Amiga multi-tasking). You can examine and modify the 68000 registers by using the REGISTER and SETREG procedures, and you can use in-line machine code by using the CODE procedure.

Non-standard Features

TDI implemented certain things a bit differently from Wirth's definition and the program must be aware of these deviations from the standard. Here are a few of these non-standard features:

a. The TDI compiler insists that open arrays should be passed by reference (that is, as VAR parameters), and this means that you can modify these arrays from within the procedure. Yet you can pass string constants as open arrays. This means that you could possibly change the passed string constant!

b. The compiler automatically removes code that could not possibly execute. For example:

```
CONST   xxx = FALSE;
.
.
.
IF xxx THEN
    WriteString('This is a test'); WriteLn
END;
```

For this piece of source code the compiler won't generate object code because it knows that xxx cannot possibly be true and the WriteString will never be executed. This feature is handy when you want to put a lot of debugging statements in your program, but you don't want the debugging code to take up any space in final version of your program.

During the debugging phase you'll declare xxx as TRUE, and in the final version you simply set it to FALSE. You must be aware, however, of a potentially dangerous effect of this feature. Suppose you have the following statement:

```
IF OpenWindow(window) AND xxx THEN ... END;
```

Well, if you have xxx declared as FALSE, then the compiler will not generate any code for this statement at all and OpenWindow will never get executed!!

c. Whenever you use the NEW and DISPOSE standard procedures in Modula-2, you have to import the ALLOCATE and DEALLOCATE procedures from the Storage modules. However, using the TDI compiler, you must also import and use the CreateHeap procedure before you can use NEW and DISPOSE, and before your program exits it must call DestroyHeap, or else the chunk of memory allocated by your program won't be accessible to the Amiga until you reboot.

d. Wirth's definition of Modula-2 allows set types to contain up to 16 elements (basically each element in a set is represented as one bit, and a set takes up 16 bits). The TDI implementation, however extended this limit to 65535 elements in a set. This is quite convenient and particularly allows you to do the following:

```
TYPE   CharSet = SET OF CHAR;
VAR    Answer : CHAR;
.
.
IF Answer IN CharSet{'A'..'D','Q'} THEN...
```

The above piece of code is quite common in Pascal programming, but impossible to do using the original definition of Modula-2.

Documentation

The TDI compiler comes with a 300-page small spiral bound manual. The first part of the manual gives instruction as to how to set up and use the compiler and linker, and also briefly discusses the standard library modules such as InOut, Streams, Storage, Strings, and MathLib0 as well as SYSTEM. The first part also discusses the various points where the TDI implementation departs from the original language definition.

The second and largest part of the manual consists of listings of the definitions modules for all the support files found in the M2 directory. If you are going to any ROM Kernel, Intuition, or AmigaDOS routines you'll be referring to this section again and again.

The last part of the manual contains an extensive item cross reference which facilitates a search for a specific identifier (constant, type, variable or procedure). For example, if you want to look up the declaration of type Window, you simply look up Window in the cross reference, find in which module it's declared (Window happens to be defined in the Intuition module), and then turn to the page that has the corresponding definition module listing.

There are some serious problems with the manual. The manual that comes with release 2.00a of the compiler is an updated version; it contains references to new modules that did not exist in the first release (such as LongInOut and RealInOut). However it contains some old information too.

For example, the definition module listing for InOut is an old listing. The actual module that comes with the compiler has some nice new procedures (such as OpenInputOutputFile, which lets you open another window and route the standard input and output to it) which are not mentioned in the manual at all. The only way you would find out about them is if you get the Developer or Commercial version and go through the definition modules on the disk. If you get the regular version... well you're out of luck.

Another problem is certain topics are discussed very briefly, and others are not discussed at all. For example, there is no mention - aside from a one-page listing of the definition module - of a module called Trapper which lets you trap and display run-time errors. The only way you would find out about it is by reading the listings for all the definition modules.

There is more information missing. I know that the implementation of type transfer functions is not complete, but there is no discussion of this fact and I have no idea exactly which type transfer functions are not implemented. The SYSTEM module exports a procedure called ExitM2. Now this sounds interesting, but there is no mention of this anywhere in the manual and I still don't know what it does.

Bugs

Yes, there are some. The compiler seems to be quite solid, but I did come across a few problems with it. The first one is the way the compiler handles the unary minus. The following code should print "0", but instead it prints "-10":

```
VAR n : INTEGER;
...
n := 5;
WriteInt(-n + 5, 1);
```

The compiler has another bug in comparing functions: Any conditional evaluation of function comparison is interpreted incorrectly. For example, suppose you have the following code:

```
IF Foo(x) > Foo(y) THEN
  WriteString('Foo(x) is greater than Foo(y)'); WriteLn
END;
```

Now suppose that Foo(x) returns 4 and Foo(y) returns 2. Well, the above IF statement is evaluated by the compiler as FALSE and the WriteString never gets executed. I've literally spent hours tracking this one down.

Another item which isn't quite a bug, but is certainly a misfeature, is the limit on the size of declared data within a module. If the total size of declared variables in a module exceeds a certain limit (let's say 35K) then the compiler will complain. However, if you spread these variables among a few modules you will be able to compile and link them all correctly.

There are more bugs in the various modules, such as Streams and other Amiga-specific modules. The nice things about Modula-2 is that as soon as a bug is found and fixed in a module, the .Lnk file for that module can be placed on various systems such as CompuServe and local bulletin boards, and everyone can download them. Compare that to C, where if you have a bug in the library file, you have to wait for a new release to get the fix.

TDI is aware of these bugs and fixes them as they are reported. I was informed by a TDI representative that a new release should be ready in about 3 months, and it will have many bug fixes, as well as the removal of the restriction on data size, and a new full-screen editor which replaces M2Error.

Benchmarks

I suppose that no compiler review is complete without some benchmarks. Well, here they are. I benchmarked the TDI compiler against the Lattice and Aztec C compilers using three programs. The program calculates primes in a very inefficient manner (I've extracted the source from the letters' column of a past issue of Amazing Computing).

This basically will test the modulo operator of the compiler. The second program is the infamous Sieve of Erathosene benchmark. The final program is the sample "Window" program found in the beginning of the Intuition manual. Please note that the Modula programs were compiled without stack and range checking. I omitted range checking because C doesn't do any range checking, and I disabled stack checking because I disabled stack checking in the Lattice benchmarks (so I could link the Lattice code without LC.Lib). The C and Modula-2 source listing for the three programs, as well as the tabulated result follow.

As always, please keep in mind benchmarks are intended to give only a rough idea as to how a compiler performs, and should be taken with a grain of salt. As you can see, the TDI compiler produced results as good as - if not better than - the Lattice C compiler. You should remember, though, that Modula-2 offers more than just good run-time performance.

Summary

Modula-2 is a new and powerful language, and is a perfect match for the Amiga. Despite its few problems, the TDI implementation is good and quite usable. I, for one, intend to stick with Modula-2 programming on the Amiga.

Suggested Reading

Should you want to learn more about Modula, here is a list of a few good books:

"Programming in Modula-2" by N. Wirth (Springer Verlag, 1985 0-387-12206-0). This is the definitive book about Modula-2. This small book is very concise, and must be read carefully. It is not the ideal book for the novice.

"Modula-2 for Pascal Programmers" by R. Gleaves (Springer Verlag, 1984). This is a good book for someone who already knows Pascal and wants to start using Modula-2.

"Modula-2: A Seafaring Guide and Shipyard Manual" by Joyce. A good introduction to Modula-2 for the novice.

"Modula-2: a Software Development Approach" by G. Ford and R. Wiener. An excellent book which covers more advanced topics in software development.

Listings

Listing 1a: Prime.Mod

```
MODULE Prime;
(*$T-*) (* turn off range checking *)
(*$S-*) (* turn off stack checking *)
FROM InOut IMPORT WriteCard, Write;

CONST
  MaxCount = 4020;
VAR
  i, n : CARDINAL;
BEGIN
  FOR n := 1 TO MaxCount DO
    i := 2;
    WHILE (i < n) AND ((n MOD i) <> 0) DO INC(i) END;
    IF i = n THEN WriteCard(n,1); Write(' '); END;
  END;
END Prime.
```

Listing 1a: Prime.C

```
/* prime.c */

#define MAX 4020

main()
{
  int i,n=0;

  while (++n<=MAX) {
    i = 1;
    while(++i<n) if(n%i==0) break;
  }
}
```

```
if(i==n) printf("%ld ",n);
}
}
```

Listing 2a: Sieve.Mod

```
MODULE Sieve;
(*$T-*) (* turn off range checking *)
(*$S-*) (* turn off stack checking *)

FROM InOut IMPORT WriteString, WriteCard, Write, WriteLn;

CONST
  size = 8190;
VAR
  flags : ARRAY[0..size] OF BOOLEAN;
  i,prime,
  k,count,
  iter : CARDINAL;

BEGIN
  WriteString('All primes from 1 to '); WriteCard(2*size+3,1);
  WriteLn;
  FOR iter := 1 TO 10 DO (* we do it 10 times !! *)
    count := 0;
    FOR i:=0 TO size DO flags[i]:=TRUE END;
    FOR i:=0 TO size DO
      IF flags[i] THEN
        prime := i+i+3;
        k:=i+prime;
        WHILE k<=size DO
          flags[k]:=FALSE;
          INC(k,prime);
        END;
        INC(count);
        (* WriteCard(prime,1); Write(' '); *)
      END;
    END;
    WriteCard(count,1); WriteString(" primes. Last one was ");
    WriteCard(prime,1); WriteLn;
  END Sieve.
```

Listing 2b: Sieve.C

```
/* Sieve.c */

#define TRUE 1
#define FALSE 0
#define size 8190

unsigned char flags[size+1];

main()
{
  int i,prime,k,count,iter;

  printf("10 ITERATIONS\n");
  for (iter=1;iter<=10;++iter) {
    count = 0;
    for (i=0;i<=size;++i) flags[i] = TRUE;
    for (i=0;i<=size;++i) {
      if (flags[i]) {
        prime = i + i + 3;
        for(k=i+prime;k<=size;k+=prime) flags[k] = FALSE;
        count++;
      }
    }
  }
}
```

```

}
printf("%ld primes\n",count);
}

```

Listing 3a: Window.mod

```

MODULE Window;
(* Based on the first sample program in the Intuition manual *)

(*$T-*) (* turn off range checking *)
(*$S-*) (* turn off stack checking *)

FROM SYSTEM IMPORT ADR, BYTE, ADDRESS, NULL;
FROM Intuition IMPORT IntuitionName, IntuitionBase, WindowFlags,
NewWindow, IDCMPFlags, IDCMPFlagSet, ScreenFlagSet,
WindowFlagSet, WindowPtr, SmartRefresh, WBenchScreen;
FROM Libraries IMPORT OpenLibrary;
FROM Windows IMPORT OpenWindow, CloseWindow;
FROM Tasks IMPORT SignalSet, Wait;
FROM InOut IMPORT WriteString, WriteLn;

CONST
    IntuitionRev = 0;
VAR
    MyWindow : NewWindow;
    WindowName : ARRAY [0..13] OF CHAR;
    wp : WindowPtr;
    Signal : SignalSet;
BEGIN
    WindowName := "Simple Window";
    (* Open intuition library *)
    IntuitionBase := OpenLibrary(IntuitionName,IntuitionRev);
    IF IntuitionBase = NULL THEN
        WriteString ("Failed to open Intuition"); WriteLn;
    ELSE (*O pened the Intuition library, so let's continue *)
        (* First, initialize the New window structure *)
        WITH MyWindow DO
            LeftEdge := 20;
            TopEdge := 20;
            Width := 300;
            Height := 100;
            DetailPen := BYTE(0);
            BlockPen := BYTE(1);
            Title := ADR(WindowName);
            Flags := WindowFlagSet {Activate, WindowClose, WindowDrag,
            WindowDepth, WindowSizing, NoCareRefresh} + SmartRefresh;
            IDCMPFlags := IDCMPFlagSet {CloseWindowFlag};
            Type := ScreenFlagSet {WBenchScreen};
            CheckMark := NULL;
            FirstGadget := NULL;;
            Screen := NULL;
            BitMap := NULL;
            MinWidth := 10;
            MinHeight := 10;
            MaxWidth := 640;
            MaxHeight := 200;
        END;
        (* Now open the window *)
        wp := OpenWindow (MyWindow);
        (* Initialize the signal Mask *)
        Signal := SignalSet{};
        (* Convert signal to a bit mask *)
        INCL(Signal,CARDINAL(wp^.UserPort^.mpSigBit));
        (* Wait for the signal *)
        Signal := Wait(Signal);
        (* Signal was received, so let's close the window and exit *)
        CloseWindow(wp^);
    END
END Window.

```

Listing 3c: Window.C

```

#include <intuition/intuition.h>

struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;

struct NewWindow NW = {
    20,20,
    300,100,
    0,1,
    CLOSEWINDOW,
    WINDOWCLOSE | SMART_REFRESH | ACTIVATE |
    WINDOWSIZING | WINDOWDRAG | WINDOWDEPTH,
    NULL,
    NULL,
    "A Simple Window",
    NULL,
    NULL,
    100,25,
    640,200,
    WBENCHSCREEN,
};

struct Window *w;

main()
{
    IntuitionBase = (struct IntuitionBase
*)OpenLibrary("intuition.library",0);
    if(IntuitionBase==NULL) exit (FALSE);

    GfxBase=(struct GfxBase *)OpenLibrary("graphics.library",0);
    if(GfxBase==NULL) exit (FALSE);

    if((w = (struct Window *)OpenWindow(&NW))==NULL) exit (FALSE);

    Wait(1<<w->UserPort->mp_SigBit);
    CloseWindow(w);
    exit (TRUE);
}

```

Benchmark Results

Notes:

1. All times are in seconds.
2. All sizes are in bytes.
3. All compiles were done on disk. All links were done in RAM.
4. 'TDI reg' refers to linking without the optimize switch
5. 'TDI opt' refers to linking with optimization
6. Lattice code was linked using BLink 5.7 and linked without LC.Lib and without stack checking, except 'primes' which required LC.Lib because of the % (modulo) operator.
7. TDI code was compiled without stack and range checking.

Primes

	TDI reg	TDI opt	Lattice	Aztec
Compile Time	31-->		30	17
Link Time	35	44	38	62
Build Time	66	75	68	79
Run Time	55	54	313	48
Obj (.Lnk) Size	376-->		260	256
Executable Size	9184	3164	13892	4780

Sieve

	TDI reg	TDI opt	Lattice	Astec
Compile Time	35-->		32	20
Link Time	36	45	26	62
Build Time	71	80	58	82
Run Time	6.9	6.5	6.5	5.3
Obj (.Lnk) Size	830-->		404	512
Executable Size	9536	3540	2592	4888

Window

	TDI reg	TDI opt	Lattice	Astec
Compile Time	92-->		68	52
Link Time	79	102	26	74
Build Time	171	194	94	126
Obj (.Lnk) Size	1354-->		636	512
Executable Size	15276	3256	2536	1928

•AC•

Amazing Computing Index Of Advertisers

Access Associates	7
Accolade, Inc.	8
Actionware- David Crane	22
Adept Software	58
Advanced Systems Design Group	57
Akron Systems	19
Ami Project	90
Anakin Research Inc.	15
Associated Computer Services	102
Benhai	85
Bethesda Softworks	52
Byte by Byte	CIV
Cardinal Software	26
Cardinal Software	82
Central Coast Software	43
Century Systems	18
Computer Systems Associates	60
Creative Solutions	88
Crystal Computer	63
Deskware	14
Digi Pix	30
Earthbound Software	95
Eastern Telecom	20
Felsina Software	56
Gimpel Software	36
Great Coverup	34
Greenthumb Software	29
Gronert Computers	35
Innovatronics	68
Interactive Analytic Node	54
Jaberware	12
Jenday Software	27
KJ Computers	99

Lattice	5
MacroWare	55
Marksman Technologies	50
Megaport Computer Center	23
Meridian Software	43,46
Metadign	2
Micro Cibernetics	63
Micro Systems Software	42
MicroIllusion	76
Microsearch	32
MicroSmiths, Inc.	48
Mimetics	44
Olamic Systems Corp.	96
Overland Laboratories, Inc.	59
Phase Four Software Distributors	94
PiM Publications, Inc.	75,73
Progressive Peripherals	CI,CIII,16
Promiga	59
R & S Data Systems	67
Scott Anthony	11
Seven Seas Software	39
Side Effects	24
Software Supermarket	58
South Park Software	61
Stacar International	101
Sunsmile Software	60
T & L Products	93
TDI Software Inc.	64
The Computer Club	56
The Memory Location	62
The Software Factory	91
Westcom Industries	31

▶ LOGISTIX ◀

IS

Spreadsheet+Timesheet+Database+Graphics+Project Management=

Successful Management!

Computer software has always been the limiting factor in your business operation. Now Progressive Peripherals & Software, Inc. has broken through this barrier to bring you a completely new concept in integrated business software. *Logistix*, the software that outweighs all other business software products!

Logistix is the planning tool designed to provide professional business people with a solution to the limitations of current spreadsheets and project management software. With time and project management built in to the same work area, you can now solve the simultaneous problems of time and money that effect your business for today and tomorrow.

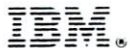
Logistix is the first and only business software program to actually integrate both time and financial functions. Face it: Time is Money. With *Logistix*, you can schedule personnel, manage shipments, plan work schedules and production flows. This allows you to monitor, project, and protect your business interests better than ever before.

Logistix offers you the four dynamic features needed to widen the horizons of success for your business. *Logistix* combines together, in one worksheet area, a large and sophisticated spreadsheet, presentation quality color graphics, database functions and, of course, powerful and flexible time and project management. Best of all, it is all designed with the business person in mind.

Logistix has built-in sideways print functions, supports over 20 international currency symbols, and offers complete support documentation, including a examples diskette. *Logistix* reads 1-2-3 files and many other file formats, so no time is lost reformatting existing data. You'll be finding business solutions faster than ever before.

Logistix is the most modern and intelligent planning tool ever designed for business people, by business people. It's loaded with features you expect to find in programs costing hundreds of dollars more. When it comes to business software, *Logistix* helps you see the future possibilities of your business.

For more information about *Logistix* call or write us today.



IBM PC/XT/AT, HP-150, Amiga and, Lotus 1-2-3 are registered trademarks of International Business Machines, Hewlett Packard Corp., Commodore Business Machines, and Lotus Development Corp. respectively. LOGISTIX is a registered trademark of GRAFOX of England and no part of this ad may be reproduced in any manner without sole written permission from Progressive Peripherals & Software, Inc.

UNLEASH THE AWESOME POWER OF THE AMIGA!™



WITH PAL SYSTEMS

- Supports Three Half Height Devices
 - Hard Disks
 - Tape Backup
 - CD ROM
- Five DMA Expansion Slots
- Battery Backed Clock/Calendar
- Whisper Fan
- Auto-Configure
- 200 Watt Power Supply
- DMA Hard Disk Controller (ST506/412)
 - Optional additional SCSI
- 100% Zorro Compatible
- 1 to 9.5 Megabytes of Fast RAM

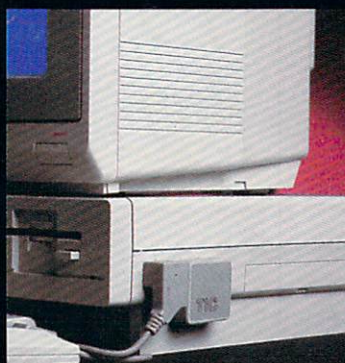
WITH PAL Jr

- One Megabyte of Fast RAM
 - DMA Hard Disk Controller
 - 20 Megabyte Hard Disk
 - Auto-Configure
 - DMA SCSI Pass-through for further expansion
- Suggested retail price only \$1495.



WITH INFOMINDER

The Information Manager. Hierarchical Database that allows you to organize and display text and graphical files, e.g. Real Estate Listings, Personnel Files, Digitized X-Rays, Geographical Maps, etc. Fully supports multi-tasking. Fast access by menu or outline. INFOMINDER will revolutionize the way you store and access both textual and graphical information. Get INFOMINDER today at the special introductory price of only \$89.95.



WITH TIC

The TIC provides your Amiga with a tiny battery backed clock/calendar that conveniently plugs into the second joystick port. The TIC's 3-year battery will maintain time even if temporarily removed from the Amiga. Change the Amiga's internal time simply by moving the displayed clock's hands with the mouse. Set your Amiga's time once and for all. It's about time for TIC. Suggested retail price only \$59.95.


BYTE by BYTE™
CORPORATION

Arboretum Plaza II
 9442 Capital of Texas Highway
 Suite 150
 Austin, TX 78759
 (512) 343-4357

AMIGA is a trademark of Commodore-Amiga, Inc.